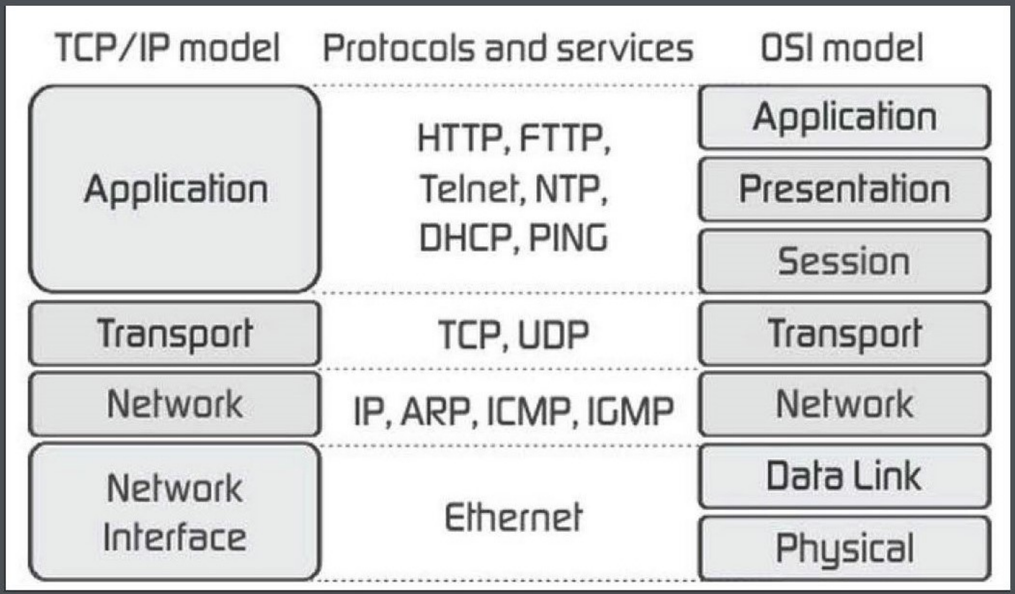
NGUYỄN TOÀN THẮNG - VŨ HUY LƯỢNG
NGUYỄN THỊ MAI PHƯƠNG - QUÁCH XUÂN TRƯỞNG

MẠNG MÁY TÍNH



TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTTP, Telnet, NTP, DHCP, PING	Application
		Presentation
		Session
Transport	TCP, UDP	Transport
Network	IP, ARP, ICMP, IGMP	Network
Network Interface	Ethernet	Data Link
		Physical

NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI

Không phần nào trong xuất bản phẩm này điược phép sao chép hay phát hành dưới bất kỳ hỉnh thức hoặc phương tiện nào nếu không được phép của Nhà xuất bân Giao thông vận tải

NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI
Số 8 phố Tăng Bạt Hổ - Phạm Đình Hổ - Hai Bà Trưng - Hà Nội
Điện thoại: 024 39423346 - 39424620 * Fax: 024 38224784
Website: www.nxbgtvt.vn

Chịu trách nhiệm xuất bản. nội dung
Giám đốc - Tổng Biên tập:
NGUYỄN MINH NHẬT
Biên tập nội dung:
VŨ VĂN BÁI
Biên tập sách điện tử:
VŨ VĂN BÁI – TRẦN ANH THU
Kỹ thuật sách điện tử:
NGÔ NHỊ LƯƠNG
Thiết kế bìa:
NGÔ NHỊ LƯƠNG

Số xác nhận đăng ký xuất bân: -2025/CXBIPH/ /GTVT
Số quyết định xuất bân: 69/QĐ-NXBGTVT ngày /2025
Nộp lưu chiểu quý II năm 2025
ISBN: 978-604-76-4154-0

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ 8
DANH MỤC HÌNH ẢNH 11
CẤU TRÚC VÀ CÁCH SỬ DỤNG GIÁO TRÌNH 16
  Cách tiếp cận vấn đề 16
  Cấu trúc giáo trình 16
  Cấu trúc bài học 17
  Ký hiệu quy ước 18
  Mục tiêu và nội dung chi tiết của các bài học 19
LỜI NÓI ĐẦU 20
BÀI 1. TỔNG QUAN VỀ MẠNG MÁY TÍNH 21
  1.1. Khái niệm chung về mạng máy tính 21
    1.1.1. Các thành phần của mạng máy tính 21
    1.1.2. Mô hình mạng 24
    1.1.3. Phân loại mạng máy tính 25
    1.1.4. Lợi ích của mạng máy tính 27
    1.1.5. Nguy cơ an ninh đối với người sử dụng mạng 28
    1.1.6. Sơ lược lịch sử phát triển của mạng máy tính 29
  1.2. Thực hành 29
    1.2.1. Kết nối máy tính với mạng không dây 29
    1.2.2. Kiểm tra kết nối mạng 30
    1.2.3. Kết nối máy tính với mạng có dây 32
    1.2.4. Lắp đặt một mạng máy tính có dây cơ bản 32
    1.2.5. Lắp đặt một mạng không dây cơ bản 34
  Bài tập và câu hỏi ôn tập 36
    Bài tập thực hành 36
    Câu hỏi lý thuyết 36
BÀI 2. TRUYỀN TÍN HIỆU TRONG MẠNG MÁY TÍNH 37
  2.1. Truyền tín hiệu trong mạng máy tính 37
    2.1.1. Khái niệm Tín hiệu trong Mạng máy tính 37

2.1.2. Các tính chất của quá trình truyền tín hiệu 40
2.1.3. Chất lượng truyền thông tin 41
2.2. Phương pháp biến đổi dữ liệu 42
2.2.1. Chuyển đổi dữ liệu trước khi truyền. 42
2.2.2. Một số sơ đồ mã hóa tín hiệu số phổ biến 44
2.3. Môi trường truyền dẫn trong mạng 47
2.3.1. Khái niệm môi trường truyền dẫn 47
2.3.2. Cáp xoắn 48
2.3.3. Cáp quang 51
2.3.4. Môi trường truyền không dây 56
2.4. Thực hành - thi công cáp xoắn 58
Bài tập và câu hỏi ôn tập 60
Bài tập thực hành 60
Câu hỏi lý thuyết 61
BÀI 3. XỬ LÝ DỮ LIỆU TRONG MẠNG MÁY TÍNH 62
3.1. Thực hành - lập trình truyền thông tin qua mạng cơ bản 62
3.2. Biến đổi dữ liệu trong mạng máy tính 65
3.2.1. Hệ nhị phân, chuỗi bit, mảng byte 66
3.2.2. Biểu diễn chuỗi bit / byte 67
3.2.3. Khái niệm endian 68
3.2.4. Biến đổi dữ liệu về mảng byte 69
3.2.5. Đóng mở gói dữ liệu 71
3.3. Thực hành - bắt và xem cấu trúc dữ liệu với Wireshark 76
3.3.1. Giới thiệu Wireshark 76
3.3.2. Sử dụng Wireshark để bắt và xem gói tin 77
Bài tập và câu hỏi ôn tập 80
Bài tập thực hành 80
Câu hỏi lý thuyết 81
BÀI 4. GIAO THỨC VÀ BỘ GIAO THỨC 82
4.1. Thực hành - Viết chương trình đơn giản cho phép truyền dữ liệu với TCP 82
4.2. Giao thức và dịch vụ truyền thông mạng 85

4.2.1. Khái niệm ... 85
4.2.2. Dịch vụ truyền thông ... 87
4.3. Bộ giao thức TCP/IP ... 90
4.3.1. Tổng quan về bộ giao thức TCP/IP ... 90
4.3.2. Giao thức UDP ... 92
4.3.3. Giao thức TCP ... 94
4.3.4. Giao thức IPv4 ... 99
4.3.5. Số cổng và địa chỉ IP ... 101
Bài tập và câu hỏi ôn tập ... 104
Bài tập thực hành ... 104
Câu hỏi lý thuyết ... 105
BÀI 5. THIẾT BỊ MẠNG ... 106
5.1. Thiết bị mạng ... 106
5.1.1. Khái niệm và đặc điểm của thiết bị mạng ... 106
5.1.2. Card giao diện mạng ... 107
5.1.3. Bộ tập trung tín hiệu ... 110
5.1.4. Bộ chuyển mạch ... 112
5.1.5. Thiết bị router ... 115
5.1.6. Hạ tầng mạng gia đình ... 118
5.2. Cấu trúc liên kết mạng ... 120
5.2.1. Khái niệm cấu trúc liên kết mạng ... 120
5.2.2. Cấu trúc hình sao ... 121
5.2.3. Cấu trúc tuyến tính ... 122
5.2.4. Cấu trúc vòng ... 123
5.2.5. Cấu trúc lưới ... 124
5.2.6. Cấu trúc liên kết và công nghệ mạng ... 125
5.3. Thực hành - mô phỏng mạng với Packet Tracer ... 126
Bài tập và câu hỏi ôn tập ... 129
Bài tập thực hành ... 129
Câu hỏi lý thuyết ... 129
BÀI 6. TRUYỀN THÔNG TIN TRONG MẠNG ... 131

6.1. Truyền thông tin trong mạng 131
    6.1.1. Địa chỉ IPv4 131
    6.1.2. Giao thức ARP 135
    6.1.3. Truyền thông tin trong mạng 136
    6.1.4. Các khái niệm liên quan đến truyền thông tin 137
6.2. Thực hành - mô phỏng quá trình truyền thông tin sử dụng Packet Tracer 139
6.3. Mô hình mạng 142
    6.3.1. Kiến trúc phân tầng 143
    6.3.2. Mô hình OSI 144
    6.3.3. Mô hình TCP/IP 145
    6.3.4. So sánh OSI với TCP/IP 146
Bài tập và câu hỏi ôn tập 148
    Bài tập thực hành 148
    Câu hỏi lý thuyết 148
BÀI 7. ỨNG DỤNG VÀ DỊCH VỤ MẠNG 150
7.1. Thực hành - Cài đặt web server 150
    7.1.1. Cài đặt IIS 150
    7.1.2. Cài đặt Apache server 153
    7.1.3. Lưu ý khi chạy server 154
7.2. Dịch vụ DNS 157
    7.2.1. Tên miền và hệ thống tên miền 157
    7.2.2. Cơ chế hoạt động của DNS 159
7.3. Thực hành - Cài đặt và sử dụng dịch vụ truyền file 160
    7.3.1. Cài đặt FileZilla client và server 161
    7.3.2. Cấu hình FileZilla server 162
    7.3.3. Kết nối FileZilla client với server 164
    7.3.4. Cấu trúc và hoạt động của hệ thống phần mềm FTP 165
Bài tập và câu hỏi ôn tập 168
    Bài tập thực hành 168
    Câu hỏi lý thuyết 168
BÀI 8. LẬP TRÌNH MẠNG 169

8.1. Xây dựng ứng dụng mạng và giao thức tầng ứng dụng 169
  8.1.1. Phát triển phần mềm mạng 169
  8.1.2. Mô hình trao đổi thông điệp trong ứng dụng mạng 171
  8.1.3. Kiến trúc ứng dụng mạng 174
8.2. Lập trình với giao thức HTTP 177
  8.2.1. Thực hành - Xây dựng web client đơn giản 177
  8.2.2. Thực hành - Xây dựng web server đơn giản 179
  8.2.3. Hệ thống web và giao thức HTTP 182
  8.2.4. Cấu trúc địa chỉ web 184
  8.2.5. Truyền dữ liệu qua HTTP 185
8.3. Lập trình ứng dụng Ping với ICMP 187
  8.3.1. Giao thức ICMP 187
  8.3.2. Thực hành: Xây dựng ứng dụng ping 189
Bài tập và câu hỏi ôn tập 191
  Bài tập thực hành 191
  Câu hỏi lý thuyết 191
BÀI 9. CÔNG NGHỆ VÀ DỊCH VỤ MẠNG HIỆN ĐẠI 193
9.1. Công nghệ mạng 193
  9.1.1. Công nghệ mạng LAN 193
  9.1.2. Công nghệ mạng WAN 195
  9.1.3. Công nghệ mạng tương lai 199
9.2. Thực hành - Sử dụng mạng riêng ảo 204
Bài tập và câu hỏi ôn tập 207
  Bài tập thực hành 207
  Câu hỏi lý thuyết 209
TÀI LIỆU THAM KHẢO 211

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Từ viết tắt	Tên tiếng Anh	Giải thích
AMQP	Advanced Message Queuing Protocol	Tên một giao thức IoT
API	Application Programming Interface	Giao diện lập trình ứng dụng
APM	Asynchronous Programming Model	Tên mô hình lập trình bất đồng bộ
ARP	Address Resolution Protocol	Giao thức phân giải địa chỉ
ASCII	American Standard Code for Information Interchange	Bảng mã chuẩn Hoa Kỳ cho trao đổi thông tin
ASP	Active Server Pages	Tên công nghệ phát triển ứng dụng web
BSD	Berkeley Software Distribution	Tên một hệ điều hành
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
CSS	Cascading Style Sheets	Ngôn ngữ định dạng giao diện web
DHCP	Dynamic Host Configuration Protocol	Giao thức cấu hình máy động
DNS	Domain Name System	Hệ thống tên miền
DTP	Data Transfer Process	Một bộ phận cấu thành của hệ thống FTP
EAP	Event-based Asynchronous Pattern	Tên một mô hình lập trình bất đồng bộ
FTP	File Transfer Protocol	Giao thức truyền file
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản cho web
HTTP	Hypertext Transport (or Transfer) Protocol	Giao thức truyền siêu bản bản cho web
ICMP	Internet Control Message Protocol	Giao thức truyền thông điệp điều khiển
IDE	Integrated Development Environment	Môi trường phát triển ứng dụng tích hợp
IGMP	Internet Group Management Protocol	Giao thức quản lý nhóm
IIS	Internet Information Services	Tên phần mềm máy chủ web
IMAP	Internet Message Access Protocol	Giao thức đồng bộ email
IoT	Internet of Things	Internet vạn vật
IP	Internet Protocol	Giao thức IP
IPC	Inter-Process Communications	Truyền thông liên tiến trình
IPTV	Internet protocol television	Giao thức TV qua Internet
ISP	Internet Service Provider	Nhà cung cấp dịch vụ Internet
JSON	JavaScript Object Notation	Tên một cấu trúc dữ liệu
LAN	Local Area Network	Mạng cục bộ
MAC	Media Access Control	Địa chỉ vật lý của thiết bị mạng

Từ viết tắt	Tên tiếng Anh	Giải thích
MLD	Multicast Listener Discovery	Tên một giao thức
MQTT	Message Queuing Telemetry Transport	Tên một giao thức trong IoT
MSA	Microsoft Message Analyzer	Phần mềm phân tích gói tin
MSS	Maximum Segment Size	Kích thước segment tối đa của TCP
MVC	Model - View - Controller	Tên một kiến trúc giao diện phần mềm
NDP	Neighbor Discovery Protocol	Giao thức phát hiện lân cận
NFS	Network File System	Hệ thống file mạng
NIC	Network Interface Card	Card giao diện mạng / card mạng
NTP	Network Time Protocol	Giao thức thời gian mạng
OSI	Open Systems Interconnection	Mô hình kết nối các hệ thống mở
P2P	Peer - to - Peer	Kiến trúc ứng dụng ngang hàng
PI	Protocol Interpreter	Thành phần thông dịch lệnh của FTP
POP3	Post Office Protocol 3	Giao thức tải email
RARP	Reverse ARP	Giao thức phân giải địa chỉ MAC về IP
RFC	Request for Comments	Tên tổ chức công bố các đặc tả giao thức
RMI	Remote Method Invocation	Gọi phương thức từ xa
RPC	Remote Procedure Call	Gọi hàm từ xa
RTP	Real-Time Transport Protocol	Giao thức truyền video thười gian thực
SIP	Session Initiation Protocol	Giao thức truyền âm thanh thời gian thực
SMTP	Simple Mail Transfer Protocol	Giao thức gửi email
SQL	Structured Query Language	Ngôn ngữ truy vấn cho các hệ quản trị cơ sở dữ liệu
TAP	Task-based Asynchronous Programming	Tên mô hình lập trình bất đồng bộ
TCP	Transmission Control Protocol	Giao thức kiểm soát truyền
TFTP	Trivial FTP	Giao thức FTP giản lược
TPL	Task Parallel Library	Tên thư viện lập trình song song của .NET
TTS	Text-to-Speech	Chuyển ăn bản thành giọng nói
UDP	User Datagram Protocol	Giao thức truyền datagram
UI	User Interface	Giao diện người dùng
URI	Uniform Resource Identifier	Định danh tài nguyên mạng
URL	Universal Resource Locator	Định vị tài nguyên mạng
UTF	Unicode Transformation Format	Định dạng chuyển đổi unicode
WCF	Windows Communications Foundation	Tên công nghệ phát triển ứng dụng mạng

Từ viết tắt	**Tên tiếng Anh**	**Giải thích**
WPF	Windows Presentation Foundation	Tên công nghệ phá triển ứng dụng windows
XAMPP	Cross-Platform Apache - MySQL - PHP - Perl	Tên bộ phần mềm máy chủ
XML	eXtensible Markup Language	Ngôn ngữ đánh dấu
YAML	Yet Another Markup Language	Tên một ngôn ngữ đánh dấu khác
ZMTP	ZeroMQ Message Transport Protocol	Tên giao thức truyền thông sử dụng trong ZeroMQ

DANH MỤC HÌNH ẢNH

Hình 1.1. Một số loại thiết bị đầu cuối 22

Hình 1.2. Một số loại môi trường truyền dẫn thông dụng 23

Hình 1.3. Một số thiết bị mạng thường gặp 24

Hình 1.4. Cách bật và kết nối máy tính với mạng không dây 30

Hình 1.5. Chạy Command Prompt từ cửa sổ Run 30

Hình 1.6. Thông tin kết nối mạng khi chạy ipconfig 31

Hình 1.7. Kết quả chạy lệnh Ping khi máy tính có kết nối Internet 31

Hình 1.8. Kết quả chạy lệnh Ping khi không có kết nối Internet 31

Hình 1.9. Sợi cáp đôi xoắn đã bấm với đầu nối 32

Hình 1.10. Cách cắm dây mạng vào máy tính và thiết bị mạng 32

Hình 1.11. Cấu hình Networkk & internet trong windows 10/11 33

Hình 1.12. Cấu hình địa chỉ mạng thủ công trong Windows 11 34

Hình 1.13. Cách thức bật Mobile hotspot trong windows 10/11 34

Hình 1.14. Cấu hình mobile hotspot 35

Hình 2.1. Tín hiệu tương tự và tín hiệu số 38

Hình 2.2. Mối quan hệ giữa băng thông và chất lượng tín hiệu số 39

Hình 2.3. Biến dạng tín hiệu do trễ 40

Hình 2.4. Tác động của nhiễu với tín hiệu số 40

Hình 2.5. Công nghệ mã hóa và điều chế 43

Hình 2.6. Định dạng tín hiệu số sau khi mã hóa 45

Hình 2.7. Tín hiệu sau điều chế thu được khi sử dụng các kỹ thuật khác nhau 46

Hình 2.8. Hình ảnh một dây cáp xoắn 48

Hình 2.9. Bộ dụng cụ thi công cáp xoắn 49

Hình 2.10. Thứ tự xếp dây trong 2 chuẩn bấm dây phổ biến 50

Hình 2.11. Một sợi cáp sau khi đã được bấm RJ45 ở hai đầu 50

Hình 2.12. Sợi cáp quang gia dụng và bó cáp quang trong công nghiệp 52

Hình 2.13. Nguyên lý truyền tín hiệu ánh sáng trong cáp quang 52

Hình 2.14. Cấu tạo một sợi cáp quang 53

Hình 2.15. Máy hàn và đồng hồ đo trong thi công cáp quang 54

Hình 2.16. Một số loại đầu nối cáp quang thường gặp.......................................55
Hình 2.17. Card mạng không dây (trái) và điểm truy cập (phải) 56
Hình 2.18. Cách dùng dao tuốt vỏ cáp...58
Hình 2.19. Cách xếp dây theo chuẩn T568B 59
Hình 2.20. Cách đẩy dây vào đầu nối RJ45 59
Hình 2.21. Cách dùng kìm bấm cáp để bấm đầu RJ45..59
Hình 2.22. Sử dụng đồng hồ kiểm tra cáp 60
Hình 3.1. Cấu trúc Solution với hai dự án UdpClient và UdpServer 62
Hình 3.2. Kết quả hoạt động của chương trình truyền thông tin qua UDP...............65
Hình 3.3. Ví dụ về cách biểu diễn một chuỗi byte 68
Hình 3.4. Trình tự đóng gói chuỗi byte ở thiết bị nguồn với TCP 72
Hình 3.5. Cấu trúc chuỗi byte header của UDP 73
Hình 3.6. Cấu trúc chuỗi byte header của IP...73
Hình 3.7. Cấu trúc chuỗi byte header của frame 74
Hình 3.8. Quy trình đóng / mở dữ liệu với TCP 75
Hình 3.9. Ý nghĩa của các phần thông tin trên cửa sổ bắt gói tin 77
Hình 3.10. Kết quả bắt gói tin UDP trong Wireshark 80
Hình 4.1. Cấu trúc solution với hai dự án, TcpClient và TcpServer 82
Hình 4.2. Kết quả hoạt động của chương trình truyền thông tin qua TCP 84
Hình 4.3. Truyền thông vật lý và truyền thông ảo 88
Hình 4.4. Cấu trúc header của gói tin UDP 93
Hình 4.5. Cấu trúc TCP header 95
Hình 4.6. Quá trình bắt tay ba bước của TCP 97
Hình 4.7. Quá trình truyền dữ liệu trong TCP 98
Hình 4.8. Cấu trúc header của IPv4..100
Hình 4.9. Cấu trúc địa chỉ IPv4...103
Hình 5.1. Hai loại card giao diện mạng gắn trong PC qua ke PCI.............................107
Hình 5.2. Cấu trúc Frame chuẩn của IEEE 802.3 108
Hình 5.3. Thông tin về card mạng khi chạy lệnh ipconfig 110
Hình 5.4. Một hub nhiều cổng của NetGear 111
Hình 5.5. Kết nối các thiết bị trong LAN với hub..111

Hình 5.6. Switch nhiều cổng của HP 112
Hình 5.7. Kết nối máy tính với switch và kết nối switch với nhau 113
Hình 5.8. Minh họa cơ chế chuyển mạch với địa chỉ MAC của switch 114
Hình 5.9. Một tủ rack với nhiều switch gắn bên trong 115
Hình 5.10. Hình ảnh một router của Cisco 115
Hình 5.11. Kết nối các mạng LAN bằng router 116
Hình 5.12. Có bao nhiêu đường đi cho dữ liệu từ client tới server? 117
Hình 5.13. Minh họa hạ tầng mạng gia đình 119
Hình 5.14. Mô hình mạng hình sao 121
Hình 5.15. Mô hình mạng tuyến tính 122
Hình 5.16. Mô hình mạng hình tròn 124
Hình 5.17. Mô hình mạng hình lưới 125
Hình 5.18. Mô hình mạng cần xây dựng 127
Hình 5.19. Cấu hình địa chỉ IP trên tab Config 127
Hình 5.20. Cấu hình địa chỉ IP bằng tiện ích IP Configuration 128
Hình 5.21. Kết quả thực hiện lệnh ping trên PC1 128
Hình 5.22. Kết quả chạy trình duyệt trên PC1 129
Hình 6.1. Cấu hình địa chỉ IP trên Windows 134
Hình 6.2. Sơ đồ mạng mô phỏng 140
Hình 6.3. Nút chuyển chế độ Simulation trên Packet Tracer 140
Hình 6.4. Giao diện Simulation Panel 140
Hình 6.5. Các bộ lọc được sử dụng 141
Hình 6.6. Kết quả mô phỏng trong Packet Tracer 141
Hình 6.7. So sánh mô hình OSI, mô hình TCP/IP và bộ giao thức TCP/IP 147
Hình 7.1. Lựa chọn Programs trong Control Panel 151
Hình 7.2. Lựa chọn Turn Windows features on or off 151
Hình 7.3. Internet Information Services với các lựa chọn mặc định 151
Hình 7.4. Giao diện IIS Manager 152
Hình 7.5. Mở trang web tự tạo trong trình duyệt 152
Hình 7.6. Lựa chọn phiên bản XAMPP trên website 153
Hình 7.7. Giao diện quản lý XAMPP Control Panel 154

Hình 7.8. Lỗi khởi động Apache khi có một web server khác đang hoạt động........154
Hình 7.9. Tắt IIS trong giao diện quản lý........155
Hình 7.10. Báo lỗi trên IIS khi có một web server khác đang hoạt động........155
Hình 7.11. Thay đổi cổng mặc định trên IIS........156
Hình 7.12. Thay đổi cấu hình Apache........156
Hình 7.13. Dòng cấu hình cổng mặc định trong file cấu hình Apache........156
Hình 7.14. Cấu trúc phân cấp của hệ thống tên miền........158
Hình 7.15 Quy trình truy vấn địa chỉ IP từ tên miền........160
Hình 7.16. Giao diện phần mềm FileZilla client........161
Hình 7.17. Khởi động FileZilla server từ giao diện quả lý của XAMPP........162
Hình 7.18. Giao diện ứng dụng quản trị của FileZilla server........162
Hình 7.19. Giao diện quản lý người dùng của FileZilla server........163
Hình 7.20. Tạo tài khoản FTP mới........163
Hình 7.21. Phân quyền người dùng........164
Hình 7.22. Thiết lập mật khẩu cho tài khoản........164
Hình 7.23. Cung cấp thông tin kết nối server trên FTP client........164
Hình 7.24. Giao diện FileZilla client sau khi kết nối tới server........165
Hình 7.25. Cấu trúc hệ thống phần mềm FTP........166
Hình 8.1. Mô hình chủ khách........174
Hình 8.2. Mô hình peer-to-peer........176
Hình 8.3. Kết quả chạy chương trình HTTP client đơn giản........178
Hình 8.4. Chương trình server cung cấp trang web cho trình duyệt........181
Hình 8.5. Mô hình hoạt động của hệ thống web........182
Hình 8.6. Cấu trúc gói tin truy vấn HTTP........183
Hình 8.7. Cấu trúc gói tin phản hồi HTTP........184
Hình 8.8. Gói tin của ICMP đặt trong gói tin IP........187
Hình 8.9. Sử dụng giao thức ICMP trong chương trình ping........188
Hình 8.10. Kết quả hoạt động của chương trình ping vừa xây dựng........190
Hình 9.1. Mối liên hệ giữa công nghệ mạng LAN và IEEE 802.3, IEEE 802.11........193
Hình 9.2. Một mạng LAN sử dụng cả công nghệ có dây và không dây........195
Hình 9.3. Sơ đồ mô hình kết nối mạng WAN sử dụng công nghệ FTTH........196

Hình 9.4. Quá trình kết nối và hoạt động của PPPoE... 198
Hình 9.5. In ấn hỗ trợ công nghệ Wifi Direct 199
Hình 9.6. Ứng dụng NFC trong đọc thông tin thẻ căn cước công dân..................... 200
Hình 9.7. Module Bluetooth 5.0 BLE MESH JDY-24M 201
Hình 9.8. Module Thu Phát Zigbee CC2530 UART 2.4G DL-22 201
Hình 9.9. Module TI CC2650 202
Hình 9.10. Truy cập Internet với VPN (trên) và không có VPN (dưới)................... 204
Hình 9.11. Thông tin về địa chỉ IP public của thiết bị... 205
Hình 9.12. Giao diện ProtonVPN client sau khi đăng nhập 206
Hình 9.13. Giao diện ProtonVPN sau khi kết nối tới server.................................... 206
Hình 9.14. Thông tin mới về kết nối Internet sau khi dùng VPN 207

CẤU TRÚC VÀ CÁCH SỬ DỤNG GIÁO TRÌNH

Cách tiếp cận vấn đề

Giáo trình Mạng máy tính được thiết kế dựa trên một cách tiếp cận hoàn toàn mới, phù hợp với định hướng đào tạo hiện đại, lấy phần mềm làm nền tảng và gắn liền lý thuyết với thực hành. Cấu trúc giáo trình không đi theo lối trình bày tuyến tính truyền thống, mà được xây dựng theo mô hình tích hợp, tạo ra mối liên kết xuyên suốt giữa các kiến thức, kỹ năng và ứng dụng thực tế.

Trước hết, giáo trình tiếp cận mạng máy tính theo **đường di chuyển của thông tin**, từ thiết bị đầu cuối, qua các tầng giao thức, đến các dịch vụ mạng và ứng dụng. Cách trình bày này giúp người học hiểu được logic vận hành tổng thể của một hệ thống mạng, từ dữ liệu thô cho đến các dịch vụ phức tạp. Song song với đó, giáo trình tổ chức nội dung theo các **khối tương tác**: phần cứng - phần mềm - giao thức - dịch vụ, từ đó khắc họa được vai trò và chức năng của từng thành phần trong một hệ thống mạng hoàn chỉnh.

Một điểm nổi bật khác của giáo trình là cách tiếp cận dạng **xoáy trôn ốc**, trong đó các khái niệm cốt lõi được giới thiệu ban đầu ở mức độ trực quan, gần gũi, sau đó được mở rộng và phân tích sâu hơn ở các bài học tiếp theo. Điều này giúp sinh viên dễ dàng tiếp thu, ghi nhớ và hình thành tư duy hệ thống, đặc biệt phù hợp với các bạn mới làm quen với môn học.

Giáo trình cũng đi **từ thực tế đến trừu tượng** - bắt đầu bằng các tình huống thực tiễn hoặc mô hình mạng quen thuộc, từ đó dẫn dắt sinh viên đến với các khái niệm lý thuyết nền tảng. Cách tổ chức này giúp người học nhận ra giá trị của kiến thức học được trong việc giải quyết các bài toán thực tế, đồng thời tăng khả năng ứng dụng liên môn.

Với cấu trúc trên, người học được khuyến khích **kết hợp học lý thuyết và thực hành** song song, thông qua các phần mô phỏng, bài tập cấu hình, và quan sát gói tin mạng. Giáo trình không chỉ cung cấp nội dung học thuật mà còn tạo điều kiện để sinh viên **rèn luyện kỹ năng tự học, tự nghiên cứu**, phát triển tư duy phân tích và khả năng khai thác mạng máy tính như một nền tảng cho các môn học và ứng dụng khác trong lĩnh vực công nghệ thông tin.

Cấu trúc giáo trình

Giáo trình này được chia thành 3 phần với 9 bài học, bao gồm:

Phần 1. Kiến thức nền tảng về mạng máy tính.

- Bài 1 - Tổng quan về mạng máy tính
- Bài 2 - Truyền tín hiệu trong mạng máy tính
- Bài 3 - Xử lý dữ liệu trong mạng máy tính

Phần 1 có mục tiêu cung cấp kiến thức nền tảng về nguyên lý của một mạng máy tính cơ bản với 2 máy tính kết nối qua môi trường truyền dẫn. Kết thúc phần 1, sinh viên sẽ hiểu được nguyên lý biến đổi dữ liệu trên thiết bị đầu cuối, cách thức biến đổi dữ liệu - tín hiệu, và cơ chế truyền tín hiệu trong mạng.

Phần 2. Giao tiếp và truyền thông trong mạng.

- Bài 4 - Giao thức và bộ giao thức
- Bài 5 - Thiết bị mạng
- Bài 6 - Truyền thông tin trong mạng

Phần 2 có mục tiêu cung cấp kiến thức chi tiết về cách vận hành của một hệ thống mạng hoàn chỉnh. Kết thúc phần 2, sinh viên sẽ hiểu được chi tiết cách thức vận hành của một mạng máy tính hoàn chỉnh, bao gồm cách hoạt động của giao thức, thiết bị mạng, và cách thông tin di chuyển trong toàn bộ hệ thống mạng.

Phần 3. Ứng dụng và công nghệ mạng.

- Bài 7 - Ứng dụng và dịch vụ mạng
- Bài 8 - Lập trình mạng
- Bài 9 - Công nghệ mạng

Phần 3 có mục tiêu cung cấp cho sinh viên các ứng dụng cụ thể của mạng máy tính. Kết thúc phần 3, sinh viên sẽ hiểu được cách ứng dụng của mạng máy tính phục vụ cho công việc và học tập trong tương lai.

Cấu trúc bài học

Mỗi bài học tương đương với 1 buổi học thực tế 5 tiết trên lớp.

Mỗi bài học được tổ chức theo cấu trúc chung như sau:

- Tóm tắt nội dung chương
- Thực hành
- Giới thiệu các vấn đề lý thuyết
- Câu hỏi ôn tập
- Bài tập thực hành

Giáo trình hướng tới giúp sinh viên tự học nên các vấn đề được trình bày một cách chi tiết, theo từng bước để sinh viên có thể tự thực hiện và nghiên cứu dưới sự hướng dẫn của giảng viên. Đo đó, cách tiếp cận của mỗi bài học là xuất phát từ một ví dụ thực hành để sinh viên có thể tự thực hiện để hình dung sơ lược về vấn đề của bài học, sau đó mới giới thiệu phần lý thuyết.

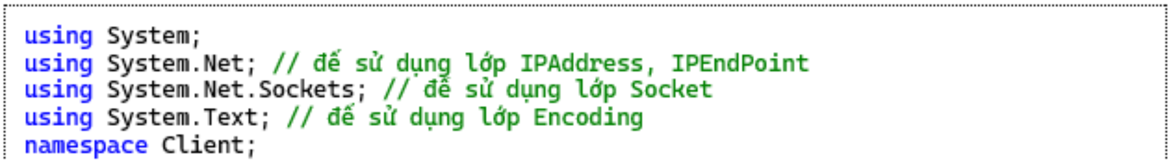
Cuối mỗi bài đều có phần câu hỏi ôn tập lý thuyết để củng cố phần kiến thức. Nếu bài học có thực hành sẽ có phần bài tập thực hành tương ứng.

Ký hiệu quy ước

Trong giáo trình này có sử dụng một số ký hiệu như sau:

- File mã nguồn trọn vẹn:

Program.cs

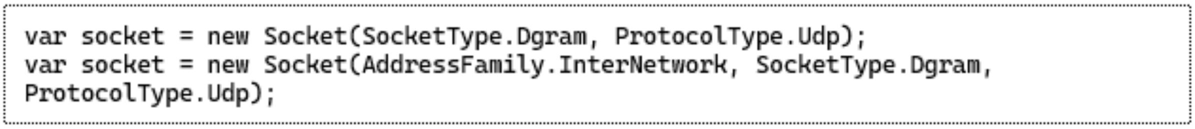


```
using System;
using System.Net; // để sử dụng lớp IPAddress, IPEndPoint
using System.Net.Sockets; // để sử dụng lớp Socket
using System.Text; // để sử dụng lớp Encoding
namespace Client;
```

File mã nguồn trọn vẹn thể hiện code với màu tiêu chuẩn của Visual Studio với tên file đặt góc trên bên phải. Khi gặp file mã nguồn trọn vẹn, sinh viên cần tìm hoặc tạo file có tên tương ứng trong dự án sau đó nhập nguyên vẹn code vào file. Nếu có nhiều dự án trong cùng một giải pháp thì tên file sẽ bao gồm cả phần tên dự án, ví dụ: Client \ Program.cs. Nếu file nằm trong thư mục thì cũng chỉ định cả tên thư mục, ví dụ: Client \ Controller \ ApiController.cs.

Thông thường file mã nguồn đều được chú thích chi tiết những dòng code quan trọng. Sinh viên cần phải đọc kỹ các chú thích này để hiểu vai trò ý nghĩa của những phần code quan trọng.

- Đoạn code minh họa:

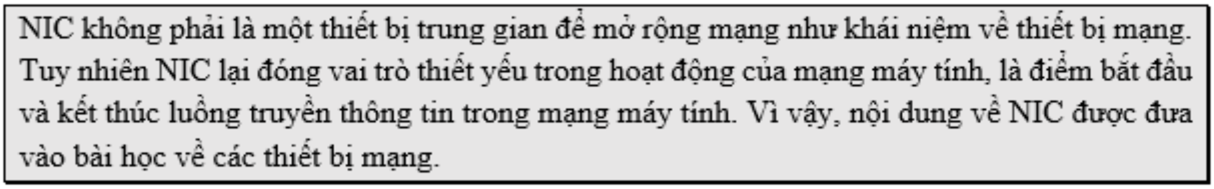


```
var socket = new Socket(SocketType.Dgram, ProtocolType.Udp);
var socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp);
```

Code minh họa là một trích đoạn code riêng rẽ, thường là một vài dòng code hoặc một phương thức riêng biệt. Code minh họa sử dụng font mono và màu mặc định của văn bản.

Code minh họa được sử dụng trong những nội dung giải thích về cách sử dụng kỹ thuật lập trình hoặc giải thích code của bài thực hành. Vì vậy code minh họa luôn phải đi kèm với nội dung khác chứ không thể sử dụng độc lập như file mã nguồn.

- Đoạn liên hệ:



NIC không phải là một thiết bị trung gian để mở rộng mạng như khái niệm về thiết bị mạng. Tuy nhiên NIC lại đóng vai trò thiết yếu trong hoạt động của mạng máy tính, là điểm bắt đầu và kết thúc luồng truyền thông tin trong mạng máy tính. Vì vậy, nội dung về NIC được đưa vào bài học về các thiết bị mạng.

Đoạn liên hệ là một đoạn văn bản mở rộng các nội dung đang trình bày được sử dụng để tránh làm rối mạch trình bày của văn bản. Đoạn liên hệ được viết với cùng font chữ như văn bản chính thức nhưng được đặt trong ô với màu nền đậm hơn.

Khi gặp đoạn liên hệ sinh viên nên đọc để có những kiến thức liên hệ khác nhưng không cần quá tập trung để tránh sao nhãng khỏi kiến thức chính đang trình bày trong phần tương ứng.

Mục tiêu và nội dung chi tiết của các bài học

Bài 1. Tổng quan về mạng máy tính

Trang bị kiến thức nền tảng về mạng như khái niệm, cấu trúc, phân loại, lợi ích, mô hình hoạt động và các thành phần chính (thiết bị đầu cuối, thiết bị mạng, môi trường truyền, giao thức). Sinh viên được hướng dẫn kết nối và kiểm tra mạng có dây/không dây.

Bài 2. Truyền tín hiệu trong mạng

Giúp sinh viên hiểu bản chất tín hiệu, mối quan hệ giữa băng thông - chất lượng truyền tin, và kỹ thuật thi công cáp. Nội dung bao gồm các loại tín hiệu, hiện tượng suy hao - nhiễu, khái niệm độ trễ và kỹ thuật thi công cáp xoắn, cáp quang.

Bài 3. Xử lý dữ liệu trong mạng

Trình bày quá trình biểu diễn, đóng gói, truyền và mở gói dữ liệu trong mạng qua lập trình và phân tích thực tế. Sinh viên học cách xử lý dữ liệu dưới dạng byte, khái niệm endian và sử dụng công cụ Wireshark.

Bài 4. Giao thức và bộ giao thức

Giới thiệu vai trò và cấu trúc bộ giao thức TCP/IP, phân biệt TCP và UDP. Sinh viên được thực hành lập trình truyền dữ liệu đơn giản với giao thức TCP.

Bài 5. Thiết bị mạng

Cung cấp kiến thức về chức năng, hoạt động của các thiết bị mạng cơ bản (NIC, hub, switch, router) và cấu trúc liên kết mạng. Thực hành thiết kế, cấu hình mạng LAN bằng Packet Tracer.

Bài 6. Truyền thông tin trong mạng

Trang bị kiến thức về địa chỉ IP, MAC, ARP, định tuyến, miền xung đột và quảng bá, đồng thời giới thiệu mô hình OSI và TCP/IP. Sinh viên mô phỏng đường đi gói tin và phân tích hoạt động truyền thông qua phần mềm.

Bài 7. Ứng dụng và dịch vụ mạng

Giúp sinh viên cài đặt, cấu hình và mô phỏng các dịch vụ Web, DNS, FTP. Qua đó hiểu rõ cách hoạt động và giao thức liên quan đến các dịch vụ mạng phổ biến.

Bài 8. Lập trình mạng

Trang bị kỹ năng xây dựng ứng dụng mạng bằng socket, mô hình client-server, peer-to-peer. Sinh viên thực hành lập trình HTTP, Ping (ICMP) và các mô hình trao đổi thông điệp.

Bài 9. Công nghệ mạng

Giới thiệu công nghệ mạng LAN, WAN và xu hướng công nghệ hiện đại như Wi-Fi Direct, BLE, Zigbee, 6LoWPAN, 5G. Sinh viên được thực hành cấu hình kết nối WAN sử dụng PPPoE.

LỜI NÓI ĐẦU

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ trên toàn cầu, mạng máy tính đã và đang đóng vai trò then chốt trong mọi hoạt động phát triển, triển khai, vận hành và bảo vệ hệ thống phần mềm. Chính vì vậy, môn học Mạng máy tính trong chương trình đào tạo 2024 được thiết kế lại theo cách tiếp cận mới, hiện đại và thực tiễn hơn, nhằm đáp ứng tốt hơn yêu cầu của thực tiễn nghề nghiệp và khả năng tích hợp với các môn học liên quan như Kiến trúc máy tính, Hệ điều hành, cũng như các chuyên ngành ứng dụng khác.

Giáo trình được xây dựng theo cách tiếp cận kết hợp: vừa theo dòng di chuyển của thông tin trong hệ thống, vừa theo khối chức năng tương tác giữa các thành phần, với cấu trúc dạng xoáy trôn ốc - từ thực tế đến trừu tượng. Mỗi bài học không chỉ cung cấp lý thuyết cốt lõi mà còn được thiết kế để gắn kết chặt chẽ với các bài thực hành mô phỏng hoặc triển khai thật, giúp sinh viên từng bước hình dung được bức tranh toàn diện về hoạt động của mạng máy tính.

Mục tiêu xuyên suốt của giáo trình là giúp sinh viên không chỉ nắm vững kiến thức về nguyên lý hoạt động của mạng, mà còn hiểu rõ mạng máy tính chính là sự phối hợp của nhiều phần mềm chạy trên nhiều thiết bị khác nhau - từ đó có thể vận dụng hiệu quả mạng vào các hoạt động phát triển và khai thác hệ thống trong môi trường thực tế. Giáo trình cũng đặc biệt chú trọng vào tính logic, tính tinh gọn, hiện đại và khả năng tự học - những yêu cầu quan trọng đối với người học trong thời đại công nghệ phát triển nhanh chóng.

Chúng tôi hy vọng rằng với cách tiếp cận mới này, sinh viên sẽ có được nền tảng vững chắc và tầm nhìn toàn diện để học tốt các môn học liên quan, cũng như tự tin bước vào thực tế nghề nghiệp trong lĩnh vực công nghệ thông tin.

Nhóm tác giả hi vọng giáo trình này sẽ là tài liệu học tập hữu ích cho sinh viên, cũng như tài liệu tham khảo cho giảng viên và những người quan tâm đến lĩnh vực mạng máy tính. Trong quá trình biên soạn, mặc dù đã có nhiều cố gắng, nhưng chắc chắn không tránh khỏi thiếu sót. Rất mong nhận được những góp ý quý báu từ bạn đọc để giáo trình ngày càng hoàn thiện hơn.

Thái Nguyên, tháng 5 năm 2025

BÀI 1. TỔNG QUAN VỀ MẠNG MÁY TÍNH

Mục tiêu của bài học

Bài học cung cấp cho sinh viên:

- Kiến thức cơ bản về mạng máy tính, định nghĩa mạng máy tính và vai trò của mạng trong việc kết nối và chia sẻ tài nguyên.
- Hiểu biết về cấu trúc hệ thống mạng máy tính và các thành phần chính bao gồm: thiết bị đầu cuối, môi trường truyền dẫn, thiết bị mạng và giao thức mạng.
- Khái niệm và phân loại thiết bị đầu cuối trong mạng máy tính.
- Kỹ năng thực hành thiết lập kết nối mạng đơn giản giữa các thiết bị đầu cuối.

Sau khi hoàn thành bài học, sinh viên có thể:

- Trình bày được khái niệm mạng máy tính và mô hình hệ thống của mạng máy tính.
- Mô tả và phân tích chức năng của các thành phần cơ bản trong mạng máy tính.
- Phân biệt được thiết bị đầu cuối với các thiết bị mạng khác.
- Nhận diện và lấy ví dụ thực tế về các thiết bị đầu cuối.
- Thực hiện được việc thiết lập một kết nối mạng cơ bản giữa hai thiết bị đầu cuối.

1.1. Khái niệm chung về mạng máy tính

Mạng máy tính là một hệ thống bao gồm ít nhất hai máy tính được kết nối với nhau để trao đổi thông tin và chia sẻ tài nguyên.

Mạng máy tính là một hệ thống (system) có nghĩa là:

- Nó bao gồm nhiều thành phần tương đối độc lập;
- Mỗi thành phần đảm nhiệm những vai trò xác định;
- Các thành phần có thể tương tác với nhau và phối hợp trong một nhiệm vụ chung.

Cụ thể, một mạng máy tính bao gồm các thiết bị đầu cuối, môi trường truyền dẫn, các thiết bị mạng, và các giao thức mạng.

1.1.1. Các thành phần của mạng máy tính

a) Thiết bị đầu cuối

Các máy tính trong mạng máy tính có tên gọi khác tổng quát hơn là thiết bị đầu cuối (end device) hoặc host. Thiết bị đầu cuối là bất kỳ thiết bị tính toán nào có thể làm phát sinh dữ liệu người dùng (điểm đầu của dữ liệu) hoặc là đích đến của dữ liệu người dùng (điểm cuối của dữ liệu). Theo đó, thiết bị đầu cuối có thể là máy tính thông thường, nhưng cũng có thể là điện thoại thông minh, thậm chí là những thiết bị gia dụng thông minh (như điều hòa, máy giặt, bình nóng lạnh, công tắc, v.v.).

Thiết bị đầu cuối có các đặc điểm quan trọng phân biệt với các thành phần khác trong mạng máy tính.

- Thiết bị đầu cuối cho phép cài đặt chương trình ứng dụng người dùng.
- Các thiết bị đầu cuối sử dụng các hệ điều hành hỗ trợ mạng.

Thiết bị đầu cuối thường được chia thành hai nhóm: máy khách và máy chủ. Máy khách dành cho người dùng cá nhân, bao gồm máy tính bàn, máy tính xách tay, điện thoại di động thông minh và máy tính bảng. Máy chủ là những máy tính đặc biệt mạnh, chạy các chương trình chuyên dụng có khả năng lưu trữ và xử lý khối lượng dữ liệu lớn.

Máy vi tính để bàn

Máy trạm

Máy tính xách tay

Raspberry Pi

Máy tính bảng

Siêu máy tính

Hình 1.1. Một số loại thiết bị đầu cuối

Việc kết nối các thiết bị đầu cuối được thực hiện theo hai mức: kết nối vật lý và kết nối logic. Kết nối vật lý cho phép tín hiệu có thể truyền qua các thiết bị. Kết nối logic cho phép dữ liệu truyền qua các thiết bị. Tín hiệu và dữ liệu được gọi chung là thông tin. Việc kết nối về mặt vật lý được thực hiện thông qua môi trường truyền dẫn và các thiết bị mạng. Việc kết nối về mặt logic được thực hiện thông qua các giao thức mạng.

b) Môi trường truyền dẫn

Môi trường truyền dẫn (transmission media) đóng vai trò quan trọng trong việc kết nối các thiết bị trong mạng máy tính. Môi trường truyền dẫn cho phép truyền tín hiệu vật lý từ một thiết bị này tới thiết bị khác.

Trong mạng máy tính, tín hiệu vật lý có thể sử dụng nhiều công nghệ khác nhau, bao gồm sóng điện từ, xung điện và xung quang học. Điều này dẫn đến sự đa dạng của môi trường truyền dẫn trong mạng máy tính.

Các môi trường truyền dẫn phổ biến bao gồm:

- Đường truyền không dây (wireless media): Môi trường truyền dẫn không dây cho phép truyền tín hiệu qua sóng điện từ.

- Các loại cáp đồng (copper cable): Các loại cáp đồng được sử dụng để truyền tín hiệu điện.
- Cáp quang (optical fiber): Cáp quang là môi trường truyền dẫn sử dụng tín hiệu ánh sáng để truyền dữ liệu.

Cáp xoắn **Cáp quang**

Hình 1.2. Một số loại môi trường truyền dẫn thông dụng

Các loại cáp dẫn tín hiệu theo hướng xác định và được gọi là **môi trường truyền dẫn có định hướng**. Trong khi đó, sóng điện từ lan truyền trong toàn bộ không gian bao quanh thiết bị phát và được gọi là **môi trường truyền dẫn không định hướng**.

c) Thiết bị mạng

Thiết bị mạng (network device) là những thành phần trung gian nằm trên đường đi của thông tin và đảm nhiệm nhiều vai trò khác nhau: kết nối thiết bị đầu cuối với môi trường truyền, kết nối các thiết bị đầu cuối với nhau, kết nối các mạng khác nhau, bảo vệ mạng, v.v.. Mục đích chính của các thiết bị mạng là đảm bảo mở rộng mạng, bao gồm mở rộng về phạm vi, chức năng, an ninh.

Đặc điểm phân biệt thiết bị mạng và thiết bị đầu cuối là:

- Thiết bị mạng không phải là nơi xuất phát và đích đến của thông tin người dùng; thiết bị mạng chỉ trung chuyển thông tin;
- Thiết bị mạng, mặc dù cũng là các máy tính chuyên dụng, nhưng không cho phép cài đặt ứng dụng người dùng.

Các thiết bị mạng thường gặp nhất là:

- Hub, repeater, switch, bridge: giúp kết nối các thiết bị đầu cuối thành một mạng phạm vi nhỏ (mạng cục bộ LAN - sẽ học ở phần tiếp theo).
- Router: giúp kết nối các mạng riêng biệt với nhau thành một mạng lớn hơn.
- Modem: giúp chuyển đổi tín hiệu từ dạng này sang dạng khác.
- Access point: giúp kết nối các thiết bị đầu cuối vào một mạng không dùng dây dẫn.
- Firewall: giúp bảo vệ mạng.

Thiết bị mạng sẽ được học chi tiết ở bài 5.

Card mạng (NIC)

Bộ chuyển mạch (switch) gia dụng

Bộ chuyển mạch (switch) công nghiệp

Bộ định tuyến (router)

Modem

Tường lửa

Hình 1.3. Một số thiết bị mạng thường gặp

d) Giao thức mạng

Giao thức mạng (network protocol) là tập hợp các quy tắc điều khiển quá trình truyền thông tin trong mạng. Với ý nghĩa này, giao thức có thể hiểu là một bản mô tả, thường tồn tại ở dạng sơ đồ và mô tả bằng ngôn ngữ tự nhiên.

Tuy nhiên, trong hệ thống mạng, các giao thức thực tế được xây dựng thành những phần mềm hoạt động trên thiết bị đầu cuối hoặc thiết bị mạng. Việc xây dựng chương trình giao thức phải tuân thủ các mô tả của giao thức.

Trong một mạng máy tính có rất nhiều giao thức cùng hoạt động. Có những giao thức hoạt động trên thiết bị đầu cuối và có thể lập trình để sử dụng. Có những giao thức hoạt động trên thiết bị mạng không thể khai thác bằng ứng dụng người dùng. Có giao thức do hệ điều hành quản lý. Có giao thức hoạt động trên phần cứng. Có những giao thức cho phép lập trình ứng dụng để khai thác nó dưới dạng dịch vụ truyền thông.

Các giao thức mạng không hoạt động độc lập mà phải tương tác với nhau để thực hiện nhiệm vụ. Một nhóm các giao thức được thiết kế để hoạt động cùng nhau tạo ra bộ giao thức (protocol suite). Hiện nay bộ giao thức chuẩn công nghiệp được sử dụng rộng rãi nhất là TCP/IP.

Giao thức mạng sẽ được học chi tiết ở chương Phần mềm và giao thức mạng.

1.1.2. Mô hình mạng

Như đã trình bày ở phần trên, mạng máy tính là một hệ thống phức tạp gồm nhiều thành phần phần cứng (như máy tính, thiết bị chuyển mạch, bộ định tuyến) và phần mềm (như giao thức truyền thông, phần mềm quản lý, dịch vụ mạng,...). Các thành phần này cần phối hợp chặt chẽ với nhau để đảm bảo việc truyền tải dữ liệu giữa các thiết bị được thực hiện một cách chính xác, nhanh chóng và an toàn. Tuy nhiên, do tính đa dạng và quy mô ngày càng lớn của mạng máy tính, việc nghiên cứu và phát triển các hệ thống mạng đòi hỏi phải có một cách tiếp cận hệ thống hóa và phân tầng hợp lý. Đó là lý do các mô hình mạng được xây dựng và sử dụng rộng rãi.

Mô hình mạng là một cấu trúc lý thuyết cho phép phân chia toàn bộ hệ thống mạng thành các tầng riêng biệt, mỗi tầng chứa các thành phần đảm nhiệm một nhóm chức năng cụ thể, và mô tả cách các thành phần ở các tầng tương tác với nhau. Việc phân tầng như vậy giúp tách biệt các vấn đề kỹ thuật, từ đó hỗ trợ cho việc thiết kế, triển khai, bảo trì và mở rộng mạng một cách linh hoạt và hiệu quả. Đặc biệt, mô hình còn là cơ sở để xây dựng các chuẩn giao thức, giúp các thiết bị và phần mềm mạng từ nhiều nhà sản xuất khác nhau có thể tương thích và hoạt động đồng bộ.

Hiện nay, có hai mô hình phổ biến nhất được sử dụng trong giảng dạy và thực tiễn là **mô hình OSI** (Open Systems Interconnection) và **mô hình TCP/IP**. Mô hình OSI do Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) phát triển, gồm 7 lớp từ lớp vật lý đến lớp ứng dụng, được thiết kế như một mô hình tham chiếu lý tưởng. Trong khi đó, mô hình TCP/IP - ra đời từ quá trình phát triển của Internet - có cấu trúc đơn giản hơn (gồm 4 hoặc 5 lớp), nhưng lại có tính thực tiễn cao hơn và được sử dụng làm nền tảng trong hầu hết các hệ thống mạng hiện đại.

Ngoài hai mô hình chính trên, một số tài liệu và hệ thống mạng cụ thể còn đưa ra các biến thể khác nhau của mô hình TCP/IP để phản ánh chi tiết hơn quá trình xử lý dữ liệu hoặc đặc điểm của hệ thống cụ thể. Dù có sự khác biệt về số lớp và cách phân loại, điểm chung của các mô hình mạng là đều nhấn mạnh đến tính phân tầng, vai trò chức năng của từng lớp, và nguyên tắc hoạt động theo hướng từ lớp thấp lên lớp cao.

Nội dung chi tiết về các mô hình mạng, cách so sánh, ánh xạ giữa các tầng, cũng như ví dụ minh họa cụ thể sẽ được trình bày trong bài học số 6 của giáo trình này. Đây là phần kiến thức nền tảng quan trọng giúp sinh viên hiểu sâu về cơ chế hoạt động của các giao thức và dịch vụ mạng trong toàn bộ hệ thống.

1.1.3. Phân loại mạng máy tính

Mạng máy tính có thể được phân loại theo nhiều tiêu chí khác nhau. Các tiêu chí phân loại mạng có thể là: môi trường truyền tín hiệu, băng thông, giao thức truyền thông, quy mô mạng, sơ đồ kết nối thiết bị, cơ chế kiểm soát luồng, kỹ thuật chuyển mạch, v.v..

Tiêu chí phân loại thông dụng nhất là phân loại theo quy mô mạng. Đây là cách phân loại gần gũi nhất và thường được đề cập đầu tiên trong những tài liệu nhập môn về mạng máy tính.

Theo cách phân loại này, mạng máy tính được chia thành các loại theo quy mô địa lý từ nhỏ đến lớn, bao gồm mạng cá nhân PAN, mạng cục bộ LAN, mạng đô thị MAN, mạng diện rộng WAN, mạng toàn cầu Internet.

a) Mạng cá nhân PAN

Mạng cá nhân PAN (Personal Area Network) cho phép các thiết bị giao tiếp qua phạm vi của một người. Một ví dụ phổ biến là mạng không dây kết nối máy tính với các thiết bị ngoại vi của nó.

Hầu hết mọi máy tính đều có màn hình, bàn phím, chuột và máy in kèm theo. Một số công ty đã cùng nhau thiết kế một mạng không dây tầm ngắn gọi là Bluetooth để kết nối các thành phần này với máy tính.

Một loại PAN hoàn toàn khác được hình thành khi một thiết bị y tế nhúng như máy điều hòa nhịp tim, máy bơm insulin hoặc máy trợ thính nói chuyện với điều khiển từ xa do người dùng vận hành.

PAN cũng có thể được xây dựng với các công nghệ khác giao tiếp trong phạm vi ngắn, chẳng hạn như RFID trên thẻ thông minh và sách thư viện.

b) Mạng cục bộ LAN

Mạng cục bộ, viết tắt theo tiếng Anh là LAN (Local Area Network), là mạng tư nhân triển khai trong phạm vi một tòa nhà, một khu vực (trường học hay cơ quan chẳng hạn) có cỡ chừng vài km.

LAN có các đặc điểm:

- Mạng LAN sử dụng một số công nghệ mạng LAN riêng biệt, bao gồm môi trường truyền dẫn và thiết bị mạng dành riêng cho LAN.
- Giới hạn về phạm vi hoạt động từ vài mét cho đến vài trăm mét.
- Vận tốc truyền dữ liệu thông thường là 100 Mbps, 1 Gbps, và gần đây là 100 Gbps.
- Thường là mạng tư nhân, tức là thuộc sở hữu của các gia đình, trường học, cơ quan, v.v..

Cần lưu ý rằng, phạm vi hoạt động và tốc độ truyền dữ liệu do công nghệ LAN quyết định. Các công nghệ LAN thường ưu tiên tốc độ truyền thay vì khoảng cách truyền.

c) Mạng diện rộng WAN

Mạng diện rộng WAN (Wide Area Network) là loại mạng có thể triển khai trên vùng địa lý rộng lớn, từ phạm vi 1 tỉnh, 1 khu vực, 1 quốc gia, hay cả lục địa.

WAN có các đặc điểm:

- Sử dụng rất nhiều công nghệ mạng khác nhau và khác biệt với các công nghệ LAN.
- Phạm vi gần như không giới hạn, từ vài trăm mét cho đến vài ngàn km.
- Tốc độ truyền dữ liệu thường thấp hơn so với LAN.
- Thường do các công ty cung cấp dịch vụ mạng / Internet sở hữu và vận hành.

Cũng chú ý rằng, các công nghệ WAN thường ưu tiên khoảng cách truyền và độ ổn định, thay vì tốc độ truyền.

d) Mạng toàn cầu Internet

Mạng toàn cầu Internet (chú ý chữ I viết hoa) có thể xem là một mạng WAN có phạm vi là toàn thế giới. Tuy nhiên, mạng Internet có một số điểm khác biệt với WAN thông thường:

- Mạng Internet có thể kết nối tất cả các mạng LAN và WAN trên toàn thế giới.
- Mạng Internet không do một tổ chức hoặc cá nhân nào sở hữu.

Cũng lưu ý rằng, đôi khi chúng ta có thể gặp thuật ngữ internet (chữ i viết thường). Đây là từ viết tắt của internetwork, thường dịch là liên mạng. Liên mạng internet là một mạng của các mạng. Hiểu một cách đơn giản, nếu có nhiều mạng khác nhau kết nối lại thành một mạng lớn hơn, đó là một liên mạng internet.

1.1.4. Lợi ích của mạng máy tính

Mạng máy tính không chỉ là nền tảng kỹ thuật quan trọng trong hệ thống công nghệ thông tin, mà còn mang lại nhiều lợi ích thiết thực cho người dùng cá nhân và tổ chức trong học tập, làm việc, giao tiếp và giải trí. Các lợi ích có thể được phân tích cụ thể như sau:

Chia sẻ tài nguyên hiệu quả: Một trong những lợi ích cơ bản của mạng máy tính là khả năng chia sẻ tài nguyên giữa nhiều người dùng. Tài nguyên ở đây có thể là phần cứng (máy in, ổ lưu trữ, máy chủ), phần mềm (ứng dụng, cơ sở dữ liệu) hoặc dữ liệu. Việc chia sẻ giúp tiết kiệm chi phí đầu tư, nâng cao hiệu quả sử dụng và tối ưu hóa hệ thống.

Trao đổi thông tin nhanh chóng: Mạng máy tính cho phép người dùng trao đổi thông tin gần như tức thời thông qua email, trò chuyện trực tuyến, hội nghị truyền hình hoặc chia sẻ tập tin. Nhờ đó, người dùng có thể phối hợp làm việc nhóm, học tập, hoặc điều hành công việc từ xa một cách hiệu quả.

Truy cập tri thức toàn cầu: Với mạng Internet, người dùng có thể dễ dàng tìm kiếm và truy cập hàng triệu tài liệu, bài giảng, nghiên cứu khoa học, khóa học trực tuyến từ khắp nơi trên thế giới. Điều này giúp nâng cao kiến thức, kỹ năng và hỗ trợ quá trình học tập, nghiên cứu liên tục và suốt đời.

Tăng cường khả năng làm việc từ xa: Mạng máy tính tạo điều kiện thuận lợi cho hình thức làm việc từ xa (remote work) hoặc học trực tuyến. Người dùng chỉ cần thiết bị có kết nối mạng là có thể tham gia vào các cuộc họp, lớp học, hoặc truy cập hệ thống cơ quan, trường học từ bất kỳ đâu.

Mở rộng khả năng tương tác xã hội: Thông qua mạng xã hội và các nền tảng kết nối số, người dùng có thể giữ liên lạc với bạn bè, gia đình, đồng nghiệp; đồng thời mở rộng mối quan hệ, hợp tác học thuật hoặc kinh doanh. Tính kết nối cao của mạng giúp rút ngắn khoảng cách địa lý và thời gian.

Giải trí đa dạng và tiện lợi: Mạng máy tính cung cấp kho nội dung giải trí phong phú như âm nhạc, phim ảnh, trò chơi, truyền hình trực tuyến. Người dùng có thể lựa chọn và trải nghiệm nội dung theo nhu cầu cá nhân mà không bị giới hạn bởi thời gian hay không gian.

Hỗ trợ cá nhân hóa dịch vụ: Các hệ thống ứng dụng trên nền tảng mạng hiện nay sử dụng dữ liệu người dùng để cung cấp dịch vụ phù hợp, từ gợi ý học liệu, đề xuất sản phẩm đến tư vấn sức khỏe. Điều này giúp trải nghiệm của người dùng trở nên thuận tiện, nhanh chóng và hiệu quả hơn.

1.1.5. Nguy cơ an ninh đối với người sử dụng mạng

Trong thời đại số, người dùng mạng máy tính - từ cá nhân đến doanh nghiệp - đều đứng trước nhiều nguy cơ mất an toàn thông tin. Dù hạ tầng mạng ngày càng hiện đại, các mối đe dọa đối với người dùng vẫn không ngừng gia tăng cả về số lượng lẫn mức độ tinh vi. Hiểu rõ những rủi ro này là bước đầu quan trọng giúp người dùng sử dụng mạng một cách an toàn và có ý thức hơn.

Một trong những nguy cơ phổ biến nhất là **lừa đảo qua mạng** (phishing). Kẻ tấn công có thể giả mạo email, trang web hoặc tin nhắn để đánh lừa người dùng cung cấp thông tin cá nhân như tài khoản ngân hàng, mật khẩu hay mã xác thực. Người dùng thường bị đánh lừa bởi giao diện quen thuộc hoặc thông tin hấp dẫn như “trúng thưởng”, “xác minh tài khoản”, hoặc “cảnh báo bảo mật”.

Một nguy cơ khác là **mất thông tin cá nhân** khi sử dụng các dịch vụ trực tuyến không an toàn. Nhiều người dùng có thói quen lưu mật khẩu trên trình duyệt, đăng nhập mạng công cộng, hoặc sử dụng phần mềm không rõ nguồn gốc. Những hành vi này tạo điều kiện cho phần mềm gián điệp (spyware) hoặc các công cụ thu thập dữ liệu đánh cắp thông tin người dùng mà họ không hề hay biết.

Ngoài ra, người dùng còn có thể là nạn nhân của **mã độc** (malware). Một số loại mã độc có thể mã hóa toàn bộ dữ liệu trong máy tính và đòi tiền chuộc (ransomware), gây hậu quả nghiêm trọng. Việc tải tệp từ nguồn không đáng tin cậy, nhấn vào các liên kết lạ hay cắm USB không kiểm tra là những hành vi tiềm ẩn rủi ro cao.

Lạm dụng mạng xã hội cũng là một vấn đề an ninh đáng lo ngại. Việc chia sẻ quá nhiều thông tin cá nhân, thói quen hàng ngày hay hình ảnh riêng tư trên mạng xã hội có thể bị kẻ xấu lợi dụng để theo dõi, đe dọa hoặc thậm chí là lừa đảo qua danh tính giả.

Cuối cùng, **mạng không dây công cộng** là một “bãi mìn” về bảo mật nếu người dùng không cẩn thận. Khi kết nối Wi-Fi miễn phí mà không có mã hóa phù hợp, toàn bộ dữ liệu truyền đi có thể bị kẻ xấu chặn lại và đọc được (tấn công kiểu "man-in-the-middle").

Do đó, để tự bảo vệ mình, người dùng cần xây dựng thói quen cảnh giác, xác thực thông tin, sử dụng mật khẩu mạnh, cập nhật phần mềm thường xuyên, và tránh chia sẻ thông tin quan trọng qua mạng không an toàn. An ninh mạng không chỉ là trách nhiệm của các chuyên gia công nghệ, mà là trách nhiệm của mỗi người sử dụng mạng.

1.1.6. Sơ lược lịch sử phát triển của mạng máy tính

Lịch sử phát triển mạng máy tính bắt đầu từ **đầu thập niên 1960** với việc hình thành khái niệm chuyển mạch gói. Năm 1961, Paul Baran (Hoa Kỳ) và Donald Davies (Anh) độc lập đề xuất kỹ thuật này nhằm thay thế cho chuyển mạch kênh truyền thống.

Đến năm **1969**, mạng **ARPANET** - mạng chuyển mạch gói đầu tiên - được xây dựng bởi ARPA (Hoa Kỳ), kết nối các máy tính tại các viện nghiên cứu. ARPANET đánh dấu mốc quan trọng đầu tiên, đặt nền móng kỹ thuật cho mạng máy tính phân tán và là tiền thân của Internet sau này.

Giai đoạn **1972-1980**, các mạng độc quyền như IBM SNA hay DECnet phát triển mạnh nhưng không tương thích lẫn nhau, thúc đẩy nhu cầu kết nối liên mạng. Giao thức **TCP/IP**, được Vinton Cerf và Robert Kahn đề xuất năm **1974**, đã giải quyết vấn đề này, tạo tiền đề cho sự ra đời của Internet.

Trong thập niên **1980**, mạng cục bộ (LAN) phát triển nhanh chóng nhờ công nghệ **Ethernet**. TCP/IP trở thành chuẩn kết nối giữa các mạng, và ARPANET dần mở rộng ra ngoài lĩnh vực quân sự, hình thành nên cấu trúc "mạng của các mạng".

Thập niên **1990** chứng kiến sự bùng nổ Internet với sự xuất hiện của **World Wide Web** (WWW, 1991) - một nền tảng giúp người dùng truy cập thông tin trực quan qua trình duyệt. Internet trở nên phổ biến rộng rãi, hỗ trợ đa dạng dịch vụ như email, tìm kiếm, thương mại điện tử và giải trí, nhờ vào sự phát triển của các nhà cung cấp dịch vụ Internet (ISP).

Bước sang thiên niên kỷ mới, Internet tiếp tục phát triển mạnh mẽ với sự ra đời của các công nghệ như **Wi-Fi, 4G/5G**, điện toán đám mây, IoT, và mạng xã hội. Internet không chỉ giới hạn trong truyền thông mà còn mở rộng ảnh hưởng tới sản xuất, y tế, giáo dục và chính phủ điện tử. Đồng thời, các vấn đề như an ninh mạng, quyền riêng tư và quản lý dữ liệu trở thành thách thức lớn trong kỷ nguyên số hóa toàn diện.

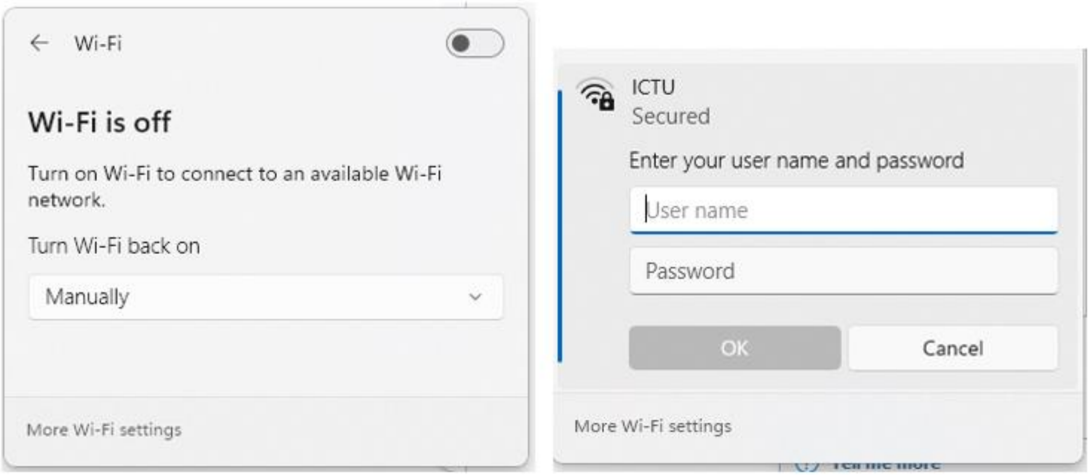
1.2. Thực hành

1.2.1. Kết nối máy tính với mạng không dây

Bước 1: Bật chức năng WiFi trên máy tính bằng nút bật/tắt ở góc trên bên phải. Ngoài ra, tại giao diện này, Windows cũng cho phép bạn tùy chọn thiết lập Wifi thời điểm tự động bật Wifi bằng lựa chọn "Turn Wifi back on" và chọn thời gian bật Wifi mặc định gồm 1 giờ, 4 giờ hoặc 1 ngày.

Bước 2: Sau khi Wifi được bật lên, danh sách các Wifi hiện khả dụng sẽ xuất hiện. Chọn tên mạng WiFi (SSID) muốn kết nối. Tiếp theo, Nhập mật khẩu (password) và tên người dùng (user name) - nếu được yêu cầu của mạng WiFi. Ngoài ra, có trường hợp chỉ cần nhập mật khẩu cho Wifi để kết nối. Mật khẩu yêu cầu sẽ có ít nhất 8 ký tự

Bước 3: Nhấp vào nút "OK/Kết nối" để kết nối tới mạng không dây đã chọn.



Wi-Fi is off

Turn on Wi-Fi to connect to an available Wi-Fi network.

Turn Wi-Fi back on

Manually

ICTU

Secured

Enter your user name and password

Hình 1.4. Cách bật và kết nối máy tính với mạng không dây

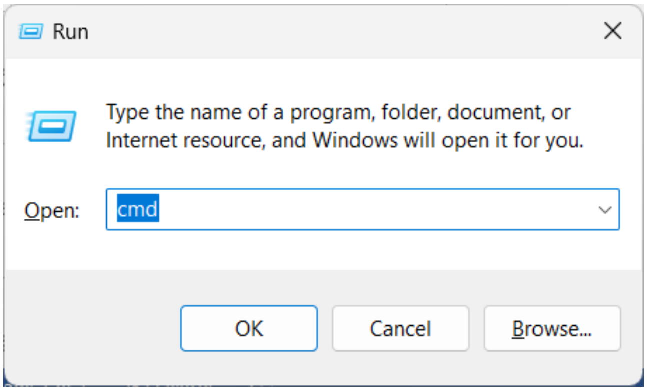
1.2.2. Kiểm tra kết nối mạng

a) Xem thông tin cấu hình

Để thực hiện các thao tác cấu hình hoặc kiểm tra kết nối mạng, chúng ta cần xem được các thông tin cấu hình mạng trên máy tính. Các hệ điều hành Windows cung cấp tiện ích ipconfig để xem thông tin cấu hình mạng của máy tính.

Để mở phần mềm ipconfig trước hết cần chạy phần mềm Command Prompt như sau:

- Bấm Windows + R để mở cửa sổ Run.
- Trong hộp văn bản gõ cmd và ấn enter để mở cửa sổ Command Prompt.

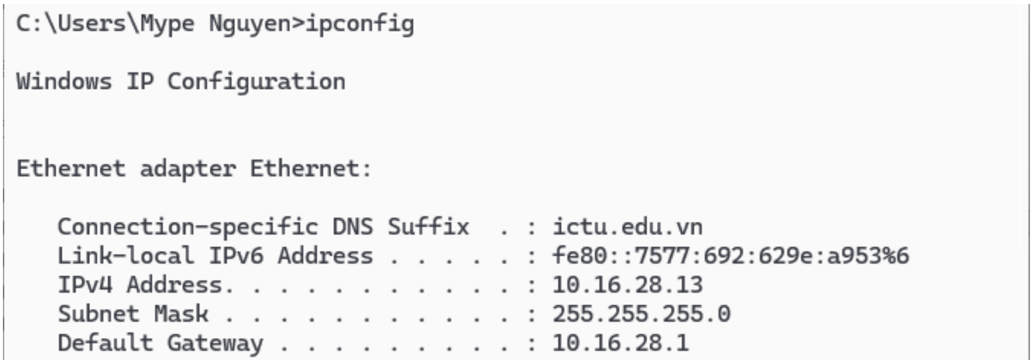


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Hình 1.5. Chạy Command Prompt từ cửa sổ Run

Trong cửa sổ Command Prompt gõ lệnh ipconfig. Lệnh này sẽ liệt kê các kết nối mạng trên máy tính và các thông tin cơ bản của kết nối mạng đó.



```
C:\Users\Mype Nguyen>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : ictu.edu.vn
   Link-local IPv6 Address . . . . . : fe80::7577:692:629e:a953%6
   IPv4 Address. . . . . . . . . . . : 10.16.28.13
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.16.28.1
```

Hình 1.6. Thông tin kết nối mạng khi chạy ipconfig

b) Kiểm tra kết nối mạng

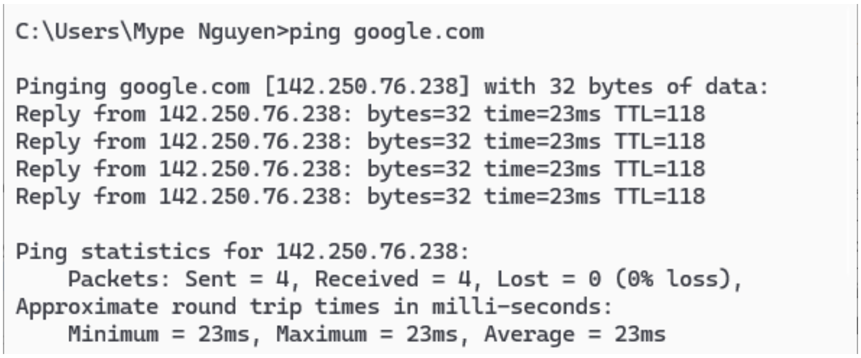
Sau khi kết nối với mạng (không dây hoặc có dây), chúng ta cần kiểm tra xem Internet đã hoạt động hay chưa. Hệ điều hành Windows cung cấp một phần mềm tiện ích có tên gọi là Ping chuyên để kiểm tra kết nối mạng.

Từ dấu nhắc lệnh của Command Prompt gõ

```
ping google.com
```

Đây là lệnh kiểm tra kết nối tới máy chủ của Google. Có thể thay google.com bắt bất kỳ địa chỉ nào khác trên Internet.

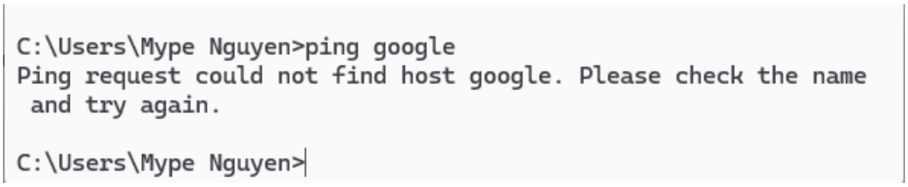
Nếu máy tính đã kết nối Internet thành công chương trình Ping sẽ trả về kết quả như sau:



```
C:\Users\Mype Nguyen>ping google.com

Pinging google.com [142.250.76.238] with 32 bytes of data:
Reply from 142.250.76.238: bytes=32 time=23ms TTL=118
Reply from 142.250.76.238: bytes=32 time=23ms TTL=118
Reply from 142.250.76.238: bytes=32 time=23ms TTL=118
Reply from 142.250.76.238: bytes=32 time=23ms TTL=118

Ping statistics for 142.250.76.238:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 23ms, Maximum = 23ms, Average = 23ms
```

Hình 1.7. Kết quả chạy lệnh Ping khi máy tính có kết nối Internet

Trong trường hợp máy tính bị mất kết nối với Internet hoặc nhập sai địa chỉ, lệnh Ping sẽ trả lại kết quả như sau:



```
C:\Users\Mype Nguyen>ping google
Ping request could not find host google. Please check the name
 and try again.

C:\Users\Mype Nguyen>
```

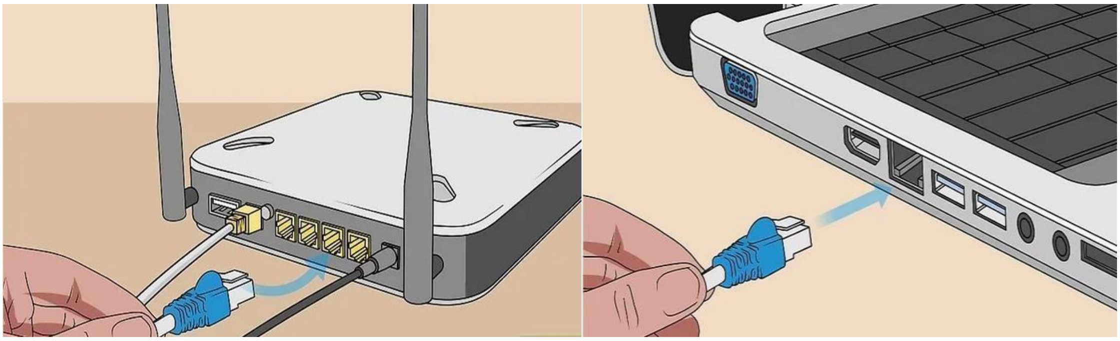
Hình 1.8. Kết quả chạy lệnh Ping khi không có kết nối Internet

1.2.3. Kết nối máy tính với mạng có dây

Để kết nối máy tính với mạng có dây chúng ta cần một sợi dây cáp mạng có tên gọi là cáp đôi xoắn (hoặc cáp UTP, cáp Ethernet, cáp LAN). Loại cáp này có thể dễ dàng mua ở các cửa hàng bán thiết bị máy tính. Chúng ta cũng có thể tự mình bấm loại cáp này sau khi học xong bài học về môi trường truyền dẫn.

Hình 1.9. Sợi cáp đôi xoắn đã bấm với đầu nối

Sợi cáp này cần cắm một đầu vào cổng LAN (còn gọi là cổng Ethernet) của máy tính, đầu còn lại cắm vào cổng LAN của thiết bị mạng do nhà cung cấp dịch vụ Internet lắp đặt. Loại thiết bị này sẽ được học chi tiết trong bài học về thiết bị mạng.



Hình 1.10. Cách cắm dây mạng vào máy tính và thiết bị mạng

Thông thường với mạng gia đình hoặc mạng cơ quan, thiết bị mạng đã được cấu hình để cung cấp thông tin cấu hình mạng tự động. Vì vậy, sau khi cắm dây, máy tính sẽ tự động thực hiện các thao tác cấu hình khác. Bạn có thể thực hiện lại các thao tác kiểm tra kết nối Internet đã học ở phần trên để xem máy tính đã có kết nối Internet hay chưa.

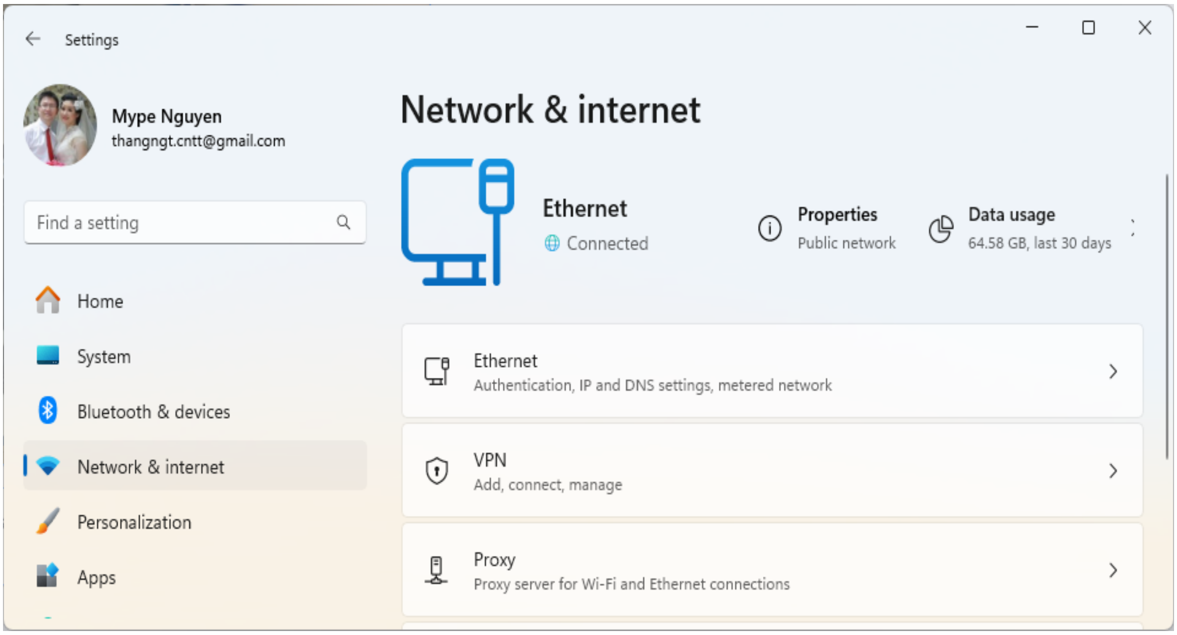
1.2.4. Lắp đặt một mạng máy tính có dây cơ bản

Bài thực hành này hướng dẫn cách lắp đặt một mạng máy tính cơ bản kết nối trực tiếp hai máy tính. Hai máy tính sau khi kết nối có thể thực hiện việc chia sẻ dữ liệu.

Một mạng máy tính cơ bản nhất chỉ bao gồm hai máy tính kết nối trực tiếp với nhau thông qua một dây cáp mạng LAN. Hãy chuẩn bị một dây cáp mạng LAN (giống như loại cáp đã sử dụng ở bài thực hành 1.2.3). Sau đó cắm mỗi đầu sợi cáp vào một máy tính.

Tuy nhiên, khi kết nối trực tiếp hai máy tính như vậy, các máy tính sẽ không thể tự động cấu hình mạng. Chúng ta sẽ phải cấu hình thủ công.

Để cấu hình thông tin kết nối mạng với máy tính sử dụng hệ điều hành Windows, người dùng có thể truy cập “Settings“ và chọn “Network & internet” để kiểm tra kết nối mạng. Thông qua trang thiết thiết lập “Network & internet”, người dùng có thể chọn Ethernet để kiểm tra và cấu hình mạng có dây.



Network & internet

Ethernet
Connected

Properties
Public network

Data usage
64.58 GB, last 30 days

Ethernet
Authentication, IP and DNS settings, metered network

VPN
Add, connect, manage

Proxy
Proxy server for Wi-Fi and Ethernet connections

Hình 1.11. Cấu hình Networkk & internet trong windows 10/11

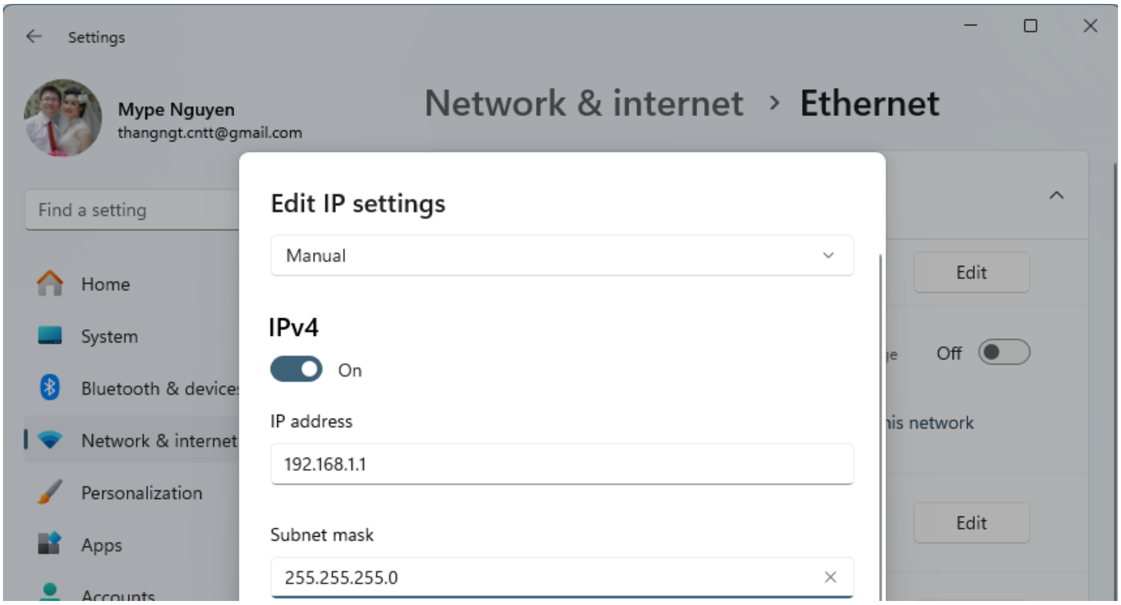
Thực hiện theo các bước dưới đây để cấu mạng giúp 2 máy tính có thể trao đổi thông tin với nhau.

Bước 1: Chuyển sang chế độ cấu hình thủ công (Manual)

Bước 2: Cấu hình địa chỉ IP (IP address) cho máy thứ nhất là 192.168.1.1, máy thứ hai là 192.168.1.2; mặt nạ mạng con (Subnet mask) cho cả hai máy là 255.255.255.0.

Sau khi lưu cấu hình, có thể sử dụng lệnh Ping để kiểm tra kết nối mạng giữa hai máy như sau:

- Trên máy thứ nhất: `ping 192.168.1.2`
- Trên máy thứ hai: `ping 192.168.1.1`


Settings
Mype Nguyen
thangngt.cntt@gmail.com
Network & internet › Ethernet
Find a setting
Home
System
Bluetooth & devices
Network & internet
Personalization
Apps
Accounts
Edit
Off
is network
Edit

Edit IP settings

Manual

IPv4

On

IP address: 192.168.1.1

Subnet mask: 255.255.255.0

Hình 1.12. Cấu hình địa chỉ mạng thủ công trong Windows 11

1.2.5. Lắp đặt một mạng không dây cơ bản

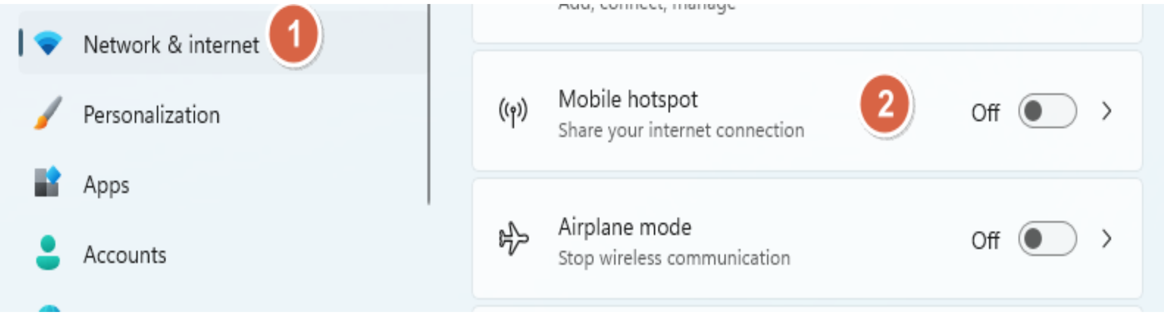
Bài thực hành hướng dẫn cách thiết lập một mạng không dây ad-hoc giữa hai máy tính laptop sử dụng tính năng Wi-Fi Hotspot trên Windows 10/11.

Để thực hiện bài thực hành này, chúng ta cần:

- 2 máy tính laptop có hỗ trợ Wi-Fi.
- Hệ điều hành Windows 10 hoặc Windows 11 đã cập nhật.
- Quyền quản trị viên (Administrator) trên cả hai máy.
- (Tùy chọn) Một số file nhỏ để kiểm tra truyền dữ liệu giữa hai máy.

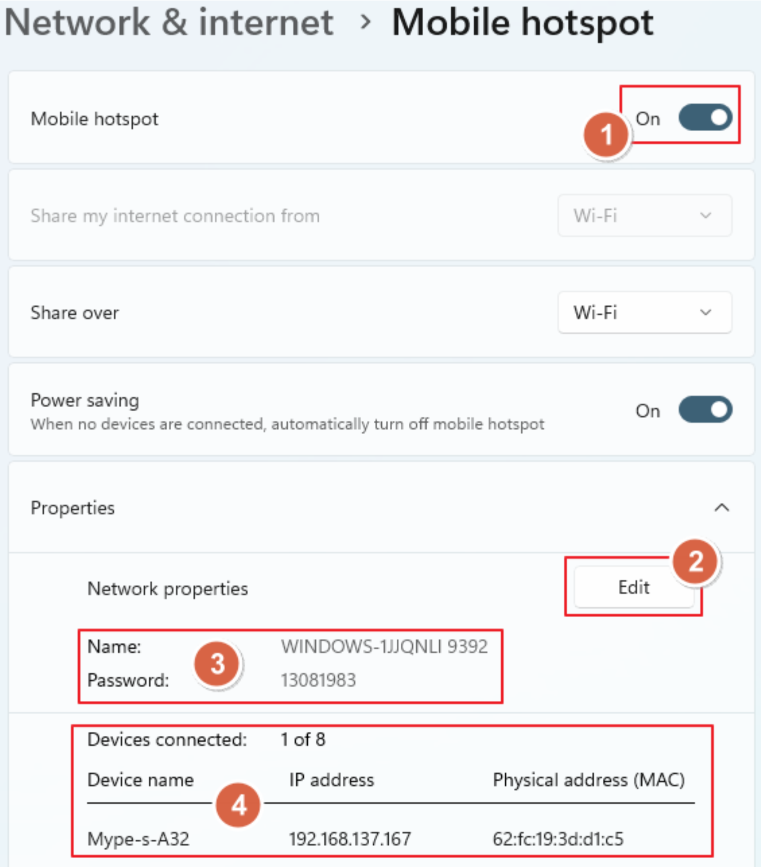
Lựa chọn một máy làm chủ và thực hiện theo các bước sau đây:

- Vào Settings > Network & Internet > Mobile Hotspot (Hình 1.13).


Network & internet
1
Personalization
Apps
Accounts
Add, connect, manage
Mobile hotspot
Share your internet connection
2
Off
Airplane mode
Stop wireless communication
Off

Hình 1.13. Cách thức bật Mobile hotspot trong windows 10/11

- Bật Mobile hotspot bằng cách kéo công tắc On (Hình 1.14).



Network & internet › Mobile hotspot

Mobile hotspot: On

Share my internet connection from: Wi-Fi

Share over: Wi-Fi

Power saving: On
When no devices are connected, automatically turn off mobile hotspot

Properties

Network properties

Name: WINDOWS-1JJQNLI 9392
Password: 13081983

Devices connected: 1 of 8

Device name	IP address	Physical address (MAC)
Mype-s-A32	192.168.137.167	62:fc:19:3d:d1:c5

Hình 1.14. Cấu hình mobile hotspot

- Tại mục Share my Internet connection from, chọn Wi-Fi.
- Tùy chọn Edit để:
    - Đặt Network name (tên mạng).
    - Đặt Password (mật khẩu).
    - Nhấn Save.

Để kết nối máy khách vào mạng Wi-Fi vừa tạo thực hiện theo các bước sau:

- Trên máy tính thứ hai (máy khách), mở Wi-Fi.
- Tìm tên mạng Wi-Fi vừa tạo.
- Nhập đúng mật khẩu đã đặt ở bước trước để kết nối.

Để kiểm tra kết nối giữa hai máy tính có thể xem bảng thông tin Properties trong mục Network & internet > Mobile hotspot. Có thể ping địa chỉ IP của máy tính khách đang kết nối để kiểm tra.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) **Thiết lập kết nối có dây:** Sử dụng cáp mạng (Ethernet), kết nối hai máy tính lại với nhau và cấu hình IP tĩnh để hai máy có thể ping được nhau.
2) **Tạo mạng không dây ad-hoc:** Tạo một mạng Wi-Fi ad-hoc giữa hai laptop bằng chức năng Mobile Hotspot hoặc thông qua thiết lập mạng thủ công trên Windows 10/11.
3) **Kiểm tra kết nối mạng:** Sử dụng lệnh `ping` trong Command Prompt để kiểm tra kết nối giữa hai thiết bị sau khi thiết lập mạng.
4) **Chia sẻ tập tin giữa hai máy:** Thiết lập chia sẻ thư mục trên một máy, sau đó truy cập thư mục đó từ máy còn lại thông qua mạng vừa tạo.
5) **Ghi lại quá trình thực hiện:** Viết một báo cáo ngắn mô tả các bước đã làm, các lỗi gặp phải (nếu có) và cách khắc phục.

Câu hỏi lý thuyết

1) Một chiếc camera giám sát IP kết nối vào hệ thống mạng qua Wi-Fi. Theo bạn, thiết bị này là thiết bị đầu cuối hay thiết bị mạng? Hãy giải thích với ví dụ thực tế.
2) Vì sao trong mạng máy tính hiện đại, kết nối logic (logical connection) lại quan trọng không kém gì kết nối vật lý? Nêu tình huống thực tế minh họa.
3) Hãy giải thích vì sao tốc độ truyền của mạng WAN lại thường chậm hơn LAN, dù công nghệ ngày càng phát triển. Có phải WAN luôn kém hơn LAN không?
4) Vì sao mạng máy tính lại được gọi là một "hệ thống"? Hãy phân tích điều này dựa trên các thành phần cơ bản của mạng.
5) Thiết bị đầu cuối và thiết bị mạng khác nhau như thế nào về chức năng và vai trò? Hãy đưa ra ví dụ minh họa cho từng loại.
6) Những thiết bị nào khác có thể được xem là thiết bị đầu cuối ngoài máy tính?
7) Một chiếc máy in Wi-Fi dùng chung trong văn phòng có được xem là thiết bị đầu cuối không? Vì sao?
8) Bạn đang kết nối laptop của mình với mạng Wi-Fi tại nhà. Những thành phần nào của mạng đang tham gia trong quá trình này? Mỗi thành phần đóng vai trò gì?
9) Tại sao nói Internet là "mạng của các mạng"? Thuật ngữ này phản ánh điều gì về cách Internet hoạt động?

BÀI 2. TRUYỀN TÍN HIỆU TRONG MẠNG MÁY TÍNH

Mục tiêu của bài học

Bài học cung cấp cho sinh viên:

- Khái niệm tín hiệu trong mạng máy tính, các đặc điểm tính chất của truyền tín hiệu trong mạng máy tính, chất lượng truyền tin trong mạng máy tính.
- Khái niệm môi trường truyền dẫn, đặc điểm tính chất của các loại cáp truyền tin thông dụng trong mạng máy tính
- Phương pháp thi công các loại cáp truyền tin thông dụng

Sau khi hoàn thành bài học, sinh viên có thể:

- Hiểu được mối quan hệ giữa chất lượng tín hiệu và băng thông, mối quan hệ giữa chất lượng tín hiệu và các yếu tố ảnh hưởng như sự suy giảm tín hiệu, méo tín hiệu, nhiễu.
- Hiểu được mối quan hệ giữa băng thông mạng và chất lượng truyền tin.
- Hiểu được mối quan hệ giữa thông lương, độ trễ và trải nghiệm người dùng
- Kỹ năng thi công cáp đôi xoắn

2.1. Truyền tín hiệu trong mạng máy tính

2.1.1. Khái niệm Tín hiệu trong Mạng máy tính

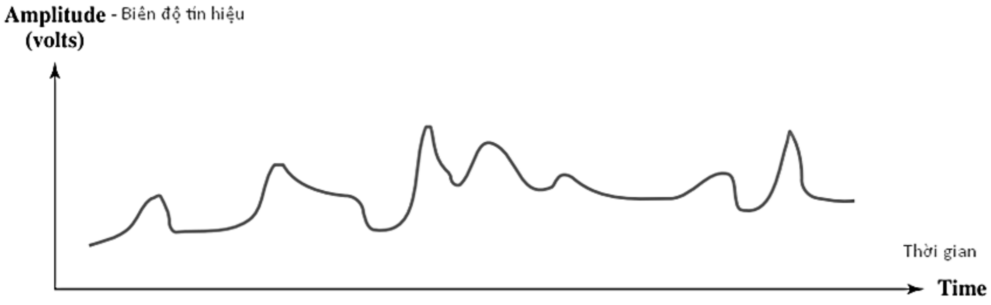
Trong mạng máy tính, **tín hiệu** là phương tiện vật lý để truyền tải dữ liệu giữa các thiết bị. Tín hiệu biểu diễn thông tin (bit 0 và bit 1) dưới dạng các biến đổi có thể đo lường được của các đại lượng vật lý như: điện áp, dòng điện, sóng điện từ, hoặc ánh sáng.

Có hai loại tín hiệu chính:

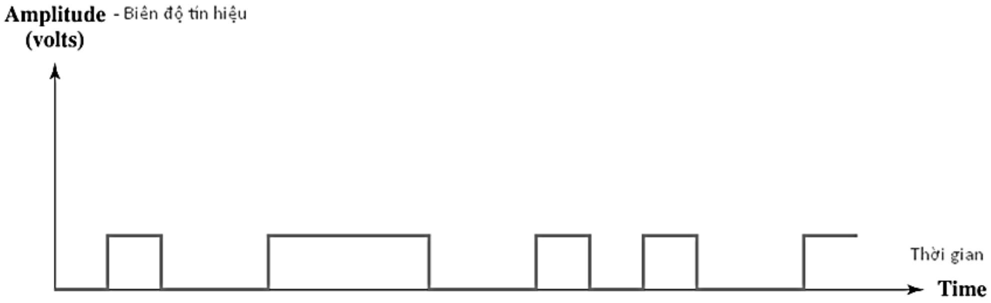
- **Tín hiệu tương tự** (analog): liên tục theo thời gian, có vô số giá trị trong một khoảng.
- **Tín hiệu số** (digital): rời rạc, chỉ biểu diễn bằng tập giá trị hữu hạn (thông thường là 0 và 1).

Tín hiệu số có đặc điểm là *cường độ* (intensity) của nó duy trì ở một mức hằng số trong một khoảng thời gian nhất định, rồi sau đó đột ngột thay đổi sang một mức hằng số khác. Nói cách khác, thay vì thay đổi liên tục như tín hiệu tương tự, tín hiệu số chỉ tồn tại ở một số mức cố định (ví dụ: mức thấp = 0, mức cao = 1). Các mức này giữ nguyên trong một khoảng thời gian nhỏ, gọi là *thời bit* (chu kỳ bit), trước khi chuyển sang mức tiếp theo. Hình 2.1 biểu diễn tín hiệu tương tự, tín hiệu số dưới dạng sóng theo thời gian.

Trong thực tế, một tín hiệu điện từ không chỉ gồm một tần số duy nhất. Thay vào đó, nó được tạo thành từ nhiều tần số khác nhau, kết hợp lại thành một sóng phức hợp. Khi chúng ta truyền một tín hiệu (ví dụ: tín hiệu số 1 hoặc 0), nó không phải là một tần số sóng đơn lẻ mượt mà.



(a) Analog - Tín hiệu tương tự



(b) Digital - Tín hiệu số

Hình 2.1. Tín hiệu tương tự và tín hiệu số

Bởi vì tín hiệu có biên độ thay đổi đột ngột (ví dụ chuyển từ 0 → 1), nên theo phân tích Fourier: Bất kỳ tín hiệu nào có cạnh sắc hoặc thay đổi đột ngột đều cần tổ hợp vô số sóng sin ở nhiều tần số khác nhau để mô tả chính xác. Các thành phần tần số cao hơn giúp tái tạo những cạnh sắc (sharp edges) trong tín hiệu số.

Ví dụ, nếu dữ liệu truyền là 1011, thì tín hiệu số sẽ có:

- Một đoạn mức cao (1),
- Một đoạn mức thấp (0),
- Một đoạn mức cao (1),
- Một đoạn mức cao (1), và giữa các đoạn này là sự chuyển đổi đột ngột, không có sự thay đổi từ từ như analog.

Để truyền một chuỗi bit dưới dạng tín hiệu từ máy gửi đến máy nhận cần một khoảng thời gian nhất định, từ đó ta có khái niệm tốc độ dữ liệu (Data rate).

Data rate (tốc độ dữ liệu) là lượng dữ liệu được truyền từ thiết bị gửi đến thiết bị nhận trong một đơn vị thời gian.

Đơn vị đo thường dùng là bit trên giây (bps - bits per second), hoặc các bội số như:

- Kbps (kilobits per second) - 1.000 bps
- Mbps (megabits per second) - 1.000.000 bps
- Gbps (gigabits per second) - 1.000.000.000 bps

Thời gian lấy mẫu (T_s) là khoảng thời gian giữa hai lần lấy mẫu liên tiếp của một tín hiệu tương tự trong quá trình chuyển đổi sang tín hiệu số, đơn vị đo là giây. **Tốc độ lấy mẫu** f_s là số lượng mẫu được thu thập từ tín hiệu tương tự trong một giây để chuyển đổi thành tín hiệu số, đơn vị đo là Hz.

$$f_s = \frac{1}{T_s}$$

Bandwidth (băng thông) là khoảng tần số mà một đường truyền hoặc thiết bị có thể sử dụng để truyền tín hiệu một cách hiệu quả. Nó cũng được hiểu là khả năng tối đa của đường truyền trong việc chuyển tải dữ liệu trong một khoảng thời gian.

> Trong phần tiếp theo chúng ta sẽ gặp một khái niệm khác, gọi là băng thông mạng (network bandwidth). Mặc dù cùng chứa từ "băng thông" nhưng chúng là hai khái niệm khác nhau.

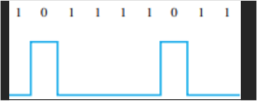
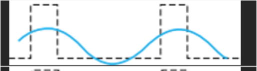
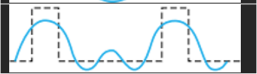
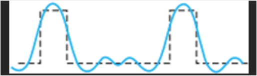
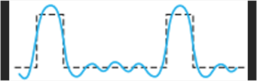
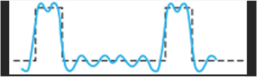
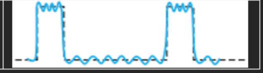
Về mặt tín hiệu, băng thông rất quan trọng quyết định chất lượng tín hiệu sau khi truyền. Dữ liệu trong Hình 2.2 cho thấy, nếu truyền một tính hiệu dạng số với tốc độ 2000 bít/giây thì băng thông để biểu diễn tín hiệu có chất lượng tốt là 2500Hz, thậm chí ở băng thông 1700Hz cũng cho ra được một biểu diễn tín hiệu tương đối tốt.

Có thể khái quát quy luật này bằng công thức:

$$B_{min} = 2*W$$

Trong đó: B_{min} (đơn vị đo là Hz) là băng thông tối thiểu cần thiết để có thể biểu diễn tốt một tín hiệu số có tốc độ bít trên giây là W.

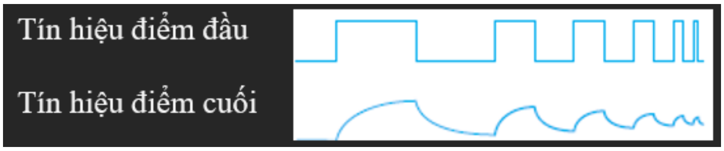
Như vậy có thể thấy rằng, có mối quan hệ giữa băng thông và chất lượng biểu diễn của tín hiệu. Băng thông càng lớn hơn B_{min} thì càng cho ra chất lượng biểu diễn tín hiệu tốt hơn.

Bít	101111011
Xung tín hiệu trước khi truyền với tốc độ 2000 bít trên giây	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 500Hz	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 900Hz	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 1300Hz	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 1700Hz	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 2500Hz	
Xung tín hiệu sau khi truyền trên đường truyền băng thông 4000Hz	

Hình 2.2. Mối quan hệ giữa băng thông và chất lượng tín hiệu số

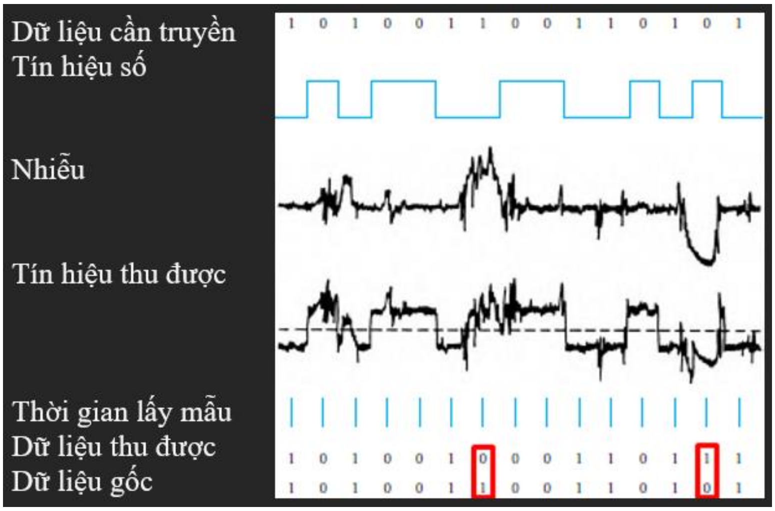
2.1.2. Các tính chất của quá trình truyền tín hiệu

Khi tín hiệu được truyền trên một môi trường bất kỳ sẽ diễn ra quá trình *suy hao tín hiệu* (Hình 2.3). **Suy hao tín hiệu** là sự giảm biên độ của tín hiệu khi nó truyền đi xa hơn trong một môi trường . Để bù suy hao, ta dùng bộ khuếch đại tín hiệu.


Tín hiệu điểm đầu
Tín hiệu điểm cuối

Hình 2.3. Biến dạng tín hiệu do trễ

Hình 2.4 biểu diễn một quá trình tín hiệu bị biến dạng từ khi tín hiệu được gửi cho đến khi tín hiệu nhận được. Hiện tượng này gọi là *biến dạng tín hiệu do trễ*. **Biến dạng do trễ** là một hiện tượng xảy ra trong quá trình truyền tín hiệu số hoặc tương tự, khi các thành phần tần số khác nhau của tín hiệu đến đích tại những thời điểm khác nhau. Điều này gây ra sự méo dạng của tín hiệu gốc, đặc biệt trong các hệ thống truyền dẫn băng rộng.


Dữ liệu cần truyền
1 0 1 0 0 1 1 0 0 1 1 0 1 0 1
Tín hiệu số
Nhiễu
Tín hiệu thu được
Thời gian lấy mẫu
Dữ liệu thu được
1 0 1 0 0 1 0 0 0 1 1 0 1 1 1
Dữ liệu gốc
1 0 1 0 0 1 1 0 0 1 1 0 1 0 1

Hình 2.4. Tác động của nhiễu với tín hiệu số

Nhiễu (noise) trong truyền tín hiệu là tín hiệu không mong muốn xen vào quá trình truyền dữ liệu, làm méo tín hiệu gốc hoặc gây lỗi trong việc giải mã thông tin tại đầu nhận. Nhiễu là tín hiệu ngẫu nhiên, không mang thông tin hữu ích, có thể xuất hiện ở bất kỳ giai đoạn nào: khi phát, trong quá trình truyền, hoặc khi thu tín hiệu. Nhiễu một trong những nguyên nhân chính gây giảm chất lượng truyền thông. Hình 2.3 mô tả quá trình tác động của nhiễu với tín hiệu số.

Nhiễu có thể được phân ra thành:

- **Nhiễu nhiệt**: là một loại nhiễu điện ngẫu nhiên sinh ra do chuyển động nhiệt của các electron trong vật dẫn (như dây đồng, điện trở, mạch điện tử). Đây là một loại nhiễu tự nhiên và không thể loại bỏ hoàn toàn, luôn tồn tại trong mọi thiết bị điện tử. Để giảm anh hưởng của nhiễu nhiệt có thể: Giảm nhiệt độ hoạt động (ví dụ: làm lạnh thiết bị điện tử); Giảm băng thông truyền nếu có thể; Tăng cường tín hiệu (dùng bộ khuếch đại tín hiệu); Sử dụng kỹ thuật mã hóa và phát hiện lỗi để đảm bảo dữ liệu không bị sai lệch.

- **Nhiễu điều chế chéo**: là một loại nhiễu xảy ra khi hai hoặc nhiều tín hiệu tần số khác nhau truyền qua cùng một thiết bị phi tuyến (đầu ra không tỷ lệ tuyến tính với đầu vào - Diot, Transitor....), và tạo ra các tín hiệu mới có tần số là tổng và hiệu của các tần số ban đầu. Những tín hiệu mới này không mong muốn và có thể gây nhiễu cho các kênh truyền khác. Để giảm thiểu Nhiễu điều chế chéo có thể: Sử dụng thiết bị tuyến tính chất lượng cao (linear amplifiers); Tách biệt các tín hiệu tần số khác nhau bằng bộ lọc; Giảm công suất đầu vào để tránh làm thiết bị hoạt động trong vùng phi tuyến; Thiết kế hệ thống anten và mạch hợp lý để tránh tương tác tín hiệu.
- **Nhiễu xuyên âm**: là hiện tượng tín hiệu từ một kênh truyền bị rò rỉ sang một kênh truyền khác, gây ra nhiễu không mong muốn. Đây là một loại nhiễu phổ biến trong các hệ thống truyền thông có nhiều dây dẫn hoặc kênh truyền gần nhau, như trong cáp mạng, cáp điện thoại, mạch in (PCB) hoặc hệ thống anten. Cách giảm thiểu Nhiễu xuyên âm: Xoắn cặp dây dẫn (twisted pair) để triệt tiêu cảm ứng điện từ; Che chắn (shielding) bằng lớp kim loại hoặc vỏ bọc chống nhiễu; Tăng khoảng cách giữa các dây dẫn hoặc kênh truyền; Thiết kế mạch in hợp lý, tránh đặt các đường tín hiệu song song quá gần nhau.
- **Nhiễu xung**: là một loại nhiễu điện xảy ra đột ngột, trong thời gian rất ngắn, nhưng có biên độ lớn. Nó thường xuất hiện dưới dạng xung nhiễu mạnh và có thể gây ra lỗi nghiêm trọng trong quá trình truyền tín hiệu, đặc biệt là trong hệ thống truyền dữ liệu số. Để giảm nhiễu xung có thể: Sử dụng bộ lọc (filters) để loại bỏ xung nhiễu; Che chắn điện từ (EMI shielding) cho cáp và thiết bị; Dùng kỹ thuật mã hóa và phát hiện/sửa lỗi (Error Detection & Correction); Cách ly thiết bị nhạy cảm khỏi nguồn gây nhiễu; Ổn định nguồn điện và bảo trì thiết bị định kỳ.

2.1.3. Chất lượng truyền thông tin

Băng thông mạng (network bandwidth) là khả năng tối đa mà một mạng có thể truyền tải dữ liệu trong một khoảng thời gian nhất định. Nó thường được đo bằng số lượng bit truyền đi trong một giây (bits per second - bps)..

Các bội số của đơn vị đo băng thông mạng:

- Kbps (Kilobits per second): 1Kbps = 10^3bps = 1000 bps.
- Mbps (Megabits per second): 1Mbps = 10^3Kbps.
- Gbps (Gigabits per second): 1Gbps = 10^3Mbps.
- Tbps (Terabits per second): 1Tbps = 10^3Gbps.

Chú ý: Băng thông (bandwidth) khác với Băng thông mạng (network bandwidth)

Giá trị của băng thông mạng do công nghệ mạng quyết định. Giá trị của băng thông mạng thường là giá trị mang tính lý thuyết, có thể tính toán ra từ công nghệ sử dụng. Trên thực tế, liên kết mạng sẽ không bao giờ đạt được tốc độ này.

Thông lượng (throughput) là lượng dữ liệu thực tế được truyền qua một hệ thống mạng trong một khoảng thời gian nhất định. Nó thường được đo bằng số lượng bit hoặc byte truyền đi trong một giây. Thông lượng thường khác với băng thông lý thuyết.

Thông lượng hữu dụng (goodput) là lượng dữ liệu hữu ích được truyền thành công từ nguồn đến đích trong một khoảng thời gian nhất định. Khác với throughput, goodput chỉ tính dữ liệu thực sự có ích cho ứng dụng, không bao gồm các gói tin bị lỗi, gói tin bị trùng lặp, hoặc các gói tin điều khiển...

Độ trễ (delay) là thời gian cần thiết để một gói dữ liệu di chuyển từ nguồn đến đích. Độ trễ là một yếu tố quan trọng ảnh hưởng đến hiệu suất và trải nghiệm người dùng trong các ứng dụng mạng. Một số loại trễ trong quá trình truyền gói tin từ nguồn tới đích bao gồm:

- Trễ xử lý: Là thời gian cần để các thiết bị mạng (như router, switch) xử lý và chuyển tiếp gói dữ liệu..
- Trễ truyền lan: Là thời gian tín hiệu cần để di chuyển qua môi trường truyền dẫn. Phụ thuộc vào khoảng cách và tốc độ truyền của tín hiệu.
- Trễ truyền tin: Là thời gian cần để truyền toàn bộ gói dữ liệu lên đường truyền. Phụ thuộc vào kích thước gói và băng thông của đường truyền..
- Trễ hàng đợi: Là thời gian gói dữ liệu phải chờ trong hàng đợi trước khi được xử lý hoặc truyền đi. Phụ thuộc vào mức độ tắc nghẽn của mạng. Khi mà số lượng gói tin gửi đến vượt quá tốc độ xử lý của thiết bị mạng, những gói tin chưa kịp xử lý được đưa lên hàng đợi để xử lý sau theo nguyên tắc đến trước xử lý trước.

Độ trễ trong mạng cần càng nhỏ càng tốt. Độ trễ quá lớn khiến quá trình truyền thông không ổn định. Đây cũng là một trong các lý do khoảng cách truyền thông tin và số lượng thiết bị mạng bị giới hạn.

2.2. Phương pháp biến đổi dữ liệu

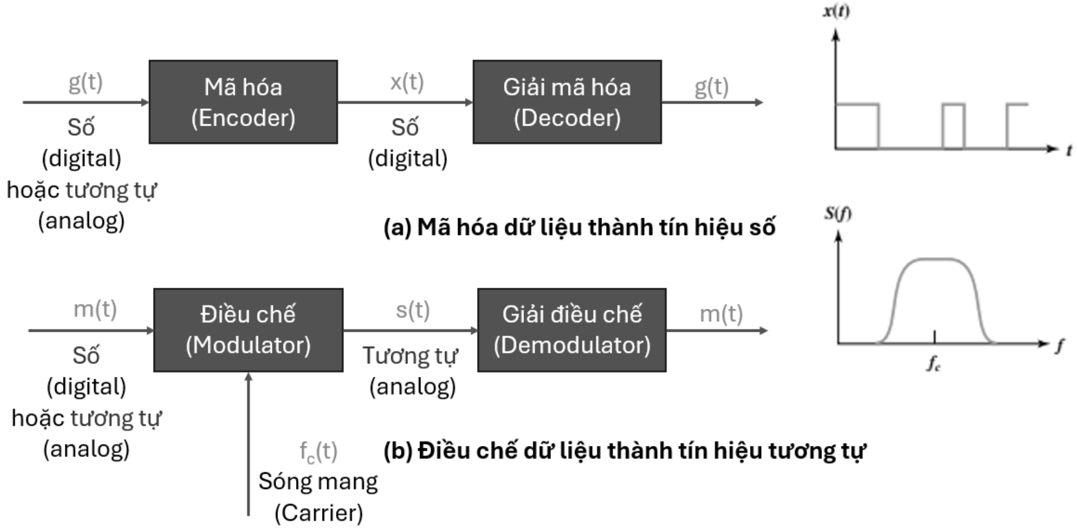
2.2.1. Chuyển đổi dữ liệu trước khi truyền.

Trước khi dữ liệu số (các bit 0 và 1) được truyền qua mạng, nó cần được chuyển đổi (encoding/modulating) thành tín hiệu vật lý phù hợp với môi trường truyền, ví dụ như điện áp (trong cáp xoắn đôi), ánh sáng (trong cáp quang), hay sóng vô tuyến (trong mạng không dây).

Lý do là vì các đường truyền vật lý không hiểu bit 0 hay 1, mà chỉ truyền được những thay đổi vật lý như mức điện áp, tần số, hoặc pha. Do đó, quá trình mã hóa tín hiệu (signal encoding) là cầu nối quan trọng để biến dữ liệu trừu tượng thành tín hiệu thực có thể di chuyển trên các phương tiện truyền thông.

Ngoài ra, việc mã hóa tín hiệu còn đóng vai trò quan trọng trong việc giảm nhiễu, tăng khả năng phát hiện lỗi, và hỗ trợ đồng bộ hóa giữa máy gửi và máy nhận.

Đối với truyền tín hiệu số, một nguồn dữ liệu g(t) có thể là dữ liệu số hoặc tương tự sẽ được mã hóa thành tín hiệu số x(t). Dạng cụ thể của x(t) phụ thuộc vào kỹ thuật mã hóa được sử dụng và được lựa chọn sao cho tối ưu hóa việc sử dụng phương tiện truyền dẫn.


g(t)
Mã hóa (Encoder)
x(t)
Giải mã hóa (Decoder)
g(t)
Số (digital) hoặc tương tự (analog)
Số (digital)
x(t)
t
S(f)
f
f_c
m(t)
Điều chế (Modulator)
s(t)
Giải điều chế (Demodulator)
m(t)
Số (digital) hoặc tương tự (analog)
Tương tự (analog)
$f_c(t)$
Sóng mang (Carrier)

(a) Mã hóa dữ liệu thành tín hiệu số

(b) Điều chế dữ liệu thành tín hiệu tương tự

Hình 2.5. Công nghệ mã hóa và điều chế

Cơ sở của truyền tín hiệu tương tự là một tín hiệu liên tục có tần số không đổi, được gọi là **tín hiệu sóng mang** (carrier signal). Tần số của tín hiệu sóng mang được lựa chọn sao cho phù hợp với phương tiện truyền dẫn đang được sử dụng. Dữ liệu có thể được truyền bằng cách *điều chế* (modulation) lên tín hiệu sóng mang. **Điều chế** là quá trình mã hóa dữ liệu nguồn lên sóng mang thông qua việc thay đổi một trong ba đặc tính cơ bản trong miền tần số: biên độ, tần số và pha.

Tín hiệu đầu vào m(t) có thể là dạng tương tự hoặc số, và được gọi là **tín hiệu điều chế** hoặc **tín hiệu cơ sở**. Kết quả của việc điều chế sóng mang được gọi là tín hiệu đã điều chế s(t). Dạng cụ thể của quá trình mã hóa được lựa chọn nhằm tối ưu hóa một đặc tính nào đó của quá trình truyền dẫn, như tiết kiệm băng thông, giảm lỗi, hoặc tăng cường khả năng chống nhiễu.

Bốn tổ hợp sơ đồ mã hóa có thể được mô tả trong Hình 2.5 đều đang được sử dụng rộng rãi. Việc lựa chọn tổ hợp nào trong một bài toán truyền thông cụ thể phụ thuộc vào nhiều yếu tố. Dưới đây là một số lý do tiêu biểu:

- Dữ liệu số, tín hiệu số: Thiết bị dùng để mã hóa dữ liệu số thành tín hiệu số thường ít phức tạp và rẻ hơn so với thiết bị điều chế từ số sang tương tự.

- Dữ liệu tương tự, tín hiệu số: Việc chuyển đổi dữ liệu tương tự sang dạng số cho phép sử dụng các thiết bị truyền và chuyển mạch số hiện đại.
- Dữ liệu số, tín hiệu tương tự: Một số phương tiện truyền, như cáp quang hoặc môi trường truyền không định hướng, chỉ có thể truyền tín hiệu tương tự (analog).
- Dữ liệu tương tự, tín hiệu tương tự: Dữ liệu tương tự dưới dạng điện có thể được truyền trực tiếp dưới dạng tín hiệu cơ sở một cách dễ dàng và chi phí thấp. Điều này thường được áp dụng cho truyền thoại qua các đường dây thoại tiêu chuẩn. Một ứng dụng phổ biến của điều chế là dịch chuyển băng thông của tín hiệu cơ sở sang một vị trí khác trên phổ tần số. Bằng cách này, nhiều tín hiệu, mỗi tín hiệu nằm ở vị trí khác nhau trên phổ, có thể chia sẻ cùng một phương tiện truyền dẫn. Đây được gọi là ghép kênh phân chia theo tần số.

Bảng 2.1. Các thuật ngữ chính trong truyền dữ liệu

Thuật ngữ	Đơn vị	Định nghĩa
Phần tử dữ liệu (Data element)	Bit	Một giá trị nhị phân đơn lẻ: 0 hoặc 1
Tốc độ dữ liệu (Data rate)	Bit trên giây (bps - bits per second)	Tốc độ mà tại đó các phần tử dữ liệu được truyền
Phần tử tín hiệu (Signal element)	Tín hiệu số: một xung điện áp có biên độ không đổi Tín hiệu tương tự: một xung có tần số, pha, và biên độ không đổi	Phần của tín hiệu chiếm khoảng thời gian ngắn nhất của một mã tín hiệu
Tốc độ tín hiệu / tốc độ điều chế (Signaling rate / Modulation rate)	Phần tử tín hiệu trên giây (baud)	Tốc độ mà tại đó các phần tử tín hiệu được truyền

2.2.2. Một số sơ đồ mã hóa tín hiệu số phổ biến

a) Dữ liệu số, tín hiệu số

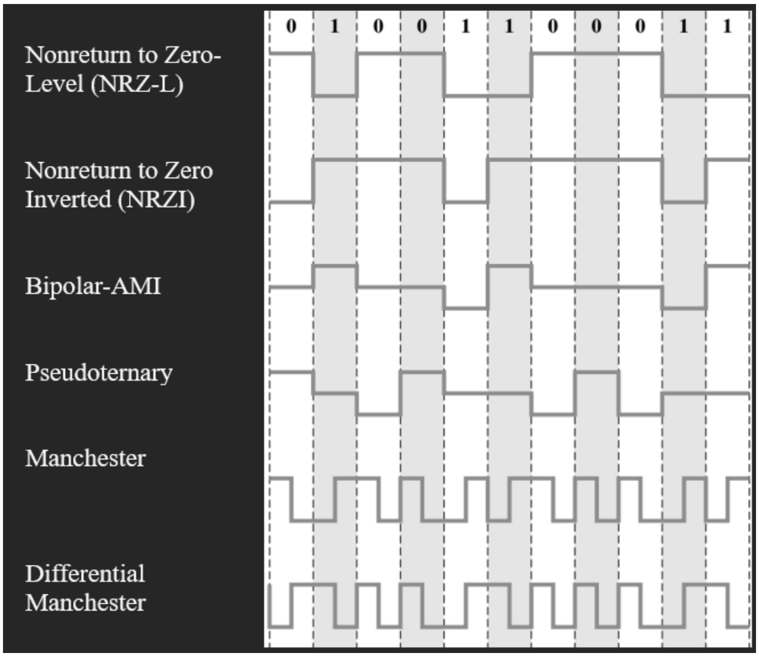
Có nhiều kỹ thuật mã hóa dữ liệu số sang tín hiệu số khác nhau, mỗi loại có ưu điểm và nhược điểm riêng. Việc lựa chọn kỹ thuật mã hóa phù hợp tùy thuộc vào môi trường truyền, yêu cầu về băng thông, khả năng đồng bộ, cũng như mức độ nhiễu có thể xảy ra. Mã hóa đóng vai trò quan trọng không chỉ trong việc truyền tín hiệu chính xác mà còn đảm bảo tính ổn định và hiệu quả của toàn bộ quá trình giao tiếp số.

Định nghĩa các định dạng mã hóa tron Hình 2.6:

- Nonreturn to Zero-Level (NRZ-L): 0 = mức cao; 1 = mức thấp
- Nonreturn to Zero Inverted (NRZI): 0 = không có chuyển trạng thái tại đầu khoảng thời gian (một chu kỳ bit); 1 = có chuyển trạng thái tại đầu khoảng thời gian

- Bipolar-AMI: 0 = không có tín hiệu đường truyền; 1 = mức dương hoặc âm, luân phiên cho các bit 1 liên tiếp
- Pseudoternary: 0 = mức dương hoặc âm, luân phiên cho các bit 0 liên tiếp; 1 = không có tín hiệu đường truyền
- Manchester: 0 = chuyển từ mức cao xuống mức thấp ở giữa khoảng thời gian; 1 = chuyển từ mức thấp lên mức cao ở giữa khoảng thời gian

Differential Manchester: Luôn có chuyển trạng thái ở giữa khoảng thời gian: 0 = có chuyển trạng thái tại đầu khoảng thời gian; 1 = không có chuyển trạng thái tại đầu khoảng thời gian



Hình 2.6. Định dạng tín hiệu số sau khi mã hóa

b) Dữ liệu số, tín hiệu tương tự

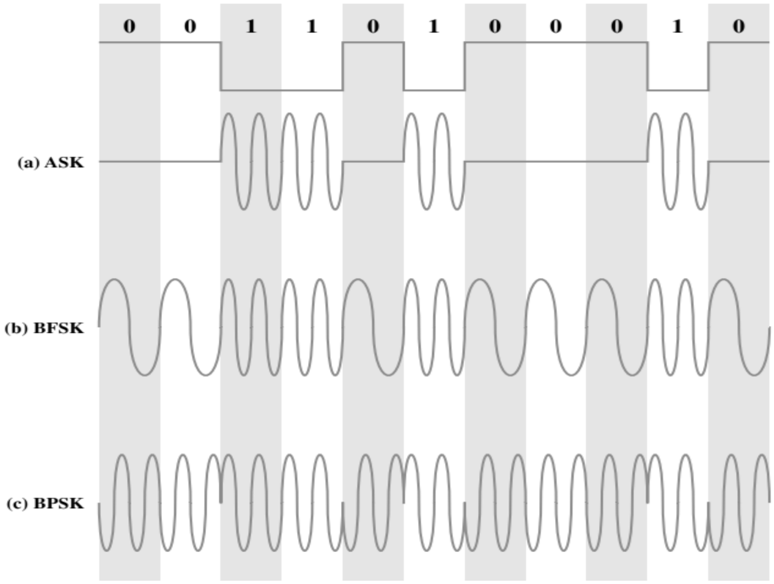
Trong trường hợp truyền dữ liệu số bằng tín hiệu tương tự, ứng dụng quen thuộc nhất của quá trình chuyển đổi này là truyền dữ liệu số qua mạng điện thoại công cộng. Mạng điện thoại được thiết kế để tiếp nhận, chuyển mạch và truyền các tín hiệu tương tự trong dải tần thoại từ khoảng 300 đến 3400 Hz. Hiện tại, mạng này không phù hợp để xử lý tín hiệu số từ các điểm thuê bao (mặc dù điều này đang dần thay đổi). Do đó, các thiết bị số được kết nối với mạng thông qua modem (bộ điều chế - giải điều chế), có chức năng chuyển đổi dữ liệu số thành tín hiệu tương tự và ngược lại.

Đối với mạng điện thoại, các modem được sử dụng để tạo ra các tín hiệu nằm trong dải tần thoại. Những kỹ thuật cơ bản tương tự cũng được áp dụng cho các modem tạo tín hiệu ở tần số cao hơn (ví dụ: vi ba - microwave).

Như đã biết điều chế liên quan đến việc tác động lên một hoặc nhiều đặc tính của sóng mang: biên độ, tần số và pha. Do đó, có ba kỹ thuật điều chế cơ bản để chuyển đổi dữ liệu số thành tín hiệu tương tự:

- Điều chế thay đổi biên độ (ASK - Amplitude Shift Keying)
- Điều chế thay đổi tần số (FSK - Frequency Shift Keying)
- Điều chế thay đổi pha (PSK - Phase Shift Keying)

Trong tất cả các trường hợp này, tín hiệu đầu ra sẽ chiếm một băng thông được định tâm tại tần số sóng mang.



Hình 2.7. Tín hiệu sau điều chế thu được khi sử dụng các kỹ thuật khác nhau

ASK (Amplitude Shift Keying): Điều chế thay đổi biên độ là kỹ thuật trong đó biên độ của sóng mang được thay đổi tùy theo giá trị dữ liệu nhị phân. Khi bit dữ liệu là 1, sóng mang được truyền với một biên độ xác định; khi bit là 0, biên độ sẽ giảm xuống thấp hơn hoặc về 0 (tắt sóng). Phương pháp này có cấu trúc đơn giản và dễ cài đặt nhưng lại dễ bị ảnh hưởng bởi nhiễu, đặc biệt là nhiễu tác động đến biên độ. ASK thường được sử dụng trong các hệ thống truyền thông yêu cầu chi phí thấp và không cần tốc độ cao.

BFSK (Binary Frequency Shift Keying là một trường hợp cụ thể của FSK): Điều chế thay đổi tần số nhị phân hoạt động bằng cách thay đổi tần số của sóng mang theo dữ liệu nhị phân. Cụ thể, một bit 1 sẽ được biểu diễn bằng một tần số f_1, và bit 0 bằng một tần số khác f_0. BFSK ít nhạy với nhiễu hơn so với ASK, do nhiễu ít khi làm thay đổi tần số tín hiệu. Tuy nhiên, BFSK cần băng thông rộng hơn để truyền tải, do phải phân biệt rõ ràng giữa hai tần số khác nhau. Kỹ thuật này thường được ứng dụng trong truyền thông không dây, các loại modem và hệ thống nhúng đơn giản.

BPSK (Binary Phase Shift Keying là một trường hợp cụ thể của PSK) - Điều chế thay đổi pha nhị phân là kỹ thuật sử dụng sự thay đổi pha của sóng mang để biểu diễn dữ liệu số. Một bit 1 thường được mã hóa bằng pha 0°, trong khi bit 0 được biểu diễn bằng pha lệch 180°. BPSK có khả năng chống nhiễu rất tốt và là kỹ thuật điều chế hiệu quả nhất trong ba phương pháp cơ bản. Tuy nhiên, việc phát hiện và giải điều chế tín hiệu yêu cầu thiết bị phức tạp hơn. BPSK thường được ứng dụng trong các hệ thống truyền thông kỹ thuật số tốc độ cao, mạng vệ tinh và thông tin quân sự.

Để hiểu sâu hơn về các kỹ thuật mã hóa, sinh viên có thể tham khảo thêm tại Chương 5 cuốn sách: “Data And Computer Communications”. Ngoài “Dữ liệu số, tín hiệu số”, “Dữ liệu số, tín hiệu tương tự” còn có “Dữ liệu tương tự, tín hiệu số”, “Dữ liệu tương tự, tín hiệu tương tự”, tuy nhiên hai cặp này ít xuất hiện trong mạng máy tính nên không được đề cập trong cuốn giáo trình này.

2.3. Môi trường truyền dẫn trong mạng

2.3.1. Khái niệm môi trường truyền dẫn

Môi trường truyền dẫn là thành phần của mạng máy tính giúp lan truyền tín hiệu giữa các thành phần phần cứng của mạng, qua đó tạo ra liên kết vật lý giữa các thành phần cứng của mạng.

Có nhiều loại tín hiệu vật lý có khả năng lan truyền, như sóng âm, các loại sóng cơ học, sóng điện từ, xung điện, xung quang học. Tuy nhiên, chỉ 3 loại tín hiệu được khai thác để tạo ra môi trường truyền dẫn trong mạng máy tính, đó là sóng điện từ, xung điện, và xung quang học. Tương ứng với các loại tín hiệu này là các loại môi trường truyền dẫn: không khí, cáp đồng, cáp quang.

Ở phía thiết bị gửi, dữ liệu (thông tin được số hóa) dưới dạng các bit được chuyển đổi thành tín hiệu rồi đưa lên đường truyền. Ở phía thiết bị nhận, tín hiệu được chuyển đổi thành dạng số. Quá trình chuyển đổi này được thực hiện bởi card mạng hoặc bởi module có nhiệm vụ tương đương trên thiết bị mạng.

Môi trường truyền dẫn giúp lan truyền tín hiệu giữa thiết bị đầu cuối và thiết bị mạng, giữa các thiết bị đầu cuối với nhau, hoặc giữa các thiết bị mạng với nhau. Không có môi trường truyền dẫn, thông tin (ở dạng tín hiệu) không để lan truyền được từ thiết bị này tới thiết bị khác, và do đó không thể tạo thành mạng máy tính.

Môi trường truyền dẫn là một bộ phận của công nghệ mạng. Ví dụ, các công nghệ mạng LAN phổ biến hiện nay như Fast Ethernet hoặc Gigabit Ethernet sử dụng cáp xoắn - là một loại cáp lõi đồng, còn công nghệ Wifi sử dụng sóng điện từ truyền trong không gian.

Môi trường truyền dẫn được chia làm hai loại: **môi trường truyền có định hướng** và **môi trường truyền không định hướng**. Môi trường truyền có định hướng dẫn tín hiệu theo đường đi xác định. Môi trường không định hướng để tín hiệu lan truyền bao phủ cả khoảng không gian. Theo phân loại này, môi trường truyền có định hướng bao gồm các loại cáp: cáp xoắn, cáp đồng trục và cáp quang. Môi trường truyền không định hướng là không gian (để sóng điện từ lan truyền).

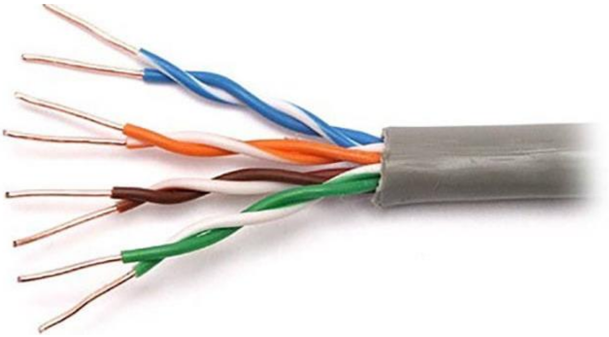
2.3.2. Cáp xoắn

Cáp xoắn là loại cáp được sử dụng phổ biến nhất trong các mạng cục bộ hiện nay.

a) Cấu tạo và phân loại cáp xoắn

Cáp xoắn (twisted pairs) là loại cáp chứa 4 cặp dây đồng, trong đó 2 dây của mỗi cặp xoắn vào nhau. Mỗi dây dẫn đều có vỏ bọc bằng nhựa. Cả 4 cặp dây cũng được đặt trong vỏ bảo vệ bằng nhựa.

Các cặp dây xoắn được đánh màu lần lượt là Cam/Trắng cam, Xanh lá/Trắng xanh lá, Xanh dương/Trắng xanh dương, Nâu/Trắng nâu. Các dây trong mỗi cặp được xoắn với nhau nhằm mục đích giảm nhiễu giữa cặp dây dẫn.



Hình 2.8. Hình ảnh một dây cáp xoắn

Cáp xoắn được chia thành một số loại khác nhau, liên quan đến *cách thức bảo vệ chống nhiễu*:

- **Cáp UTP** (Unshielded Twisted Pairs) không có lớp màng kim loại bảo vệ các cặp dây cũng như bảo vệ cả sợi cáp.
- **Cáp F/UTP** có một lớp màng kim loại bảo vệ chung cả 4 cặp dây.
- **Cáp S/FTP** có một lớp màng kim loại bảo vệ chung cả 4 cặp dây, đồng thời có thêm một lớp màng kim loại nữa cho mỗi cặp dây.

Cáp UTP là loại phổ biến và giá rẻ nhất trên thị trường. Hai loại cáp sau có giá thành cao và thường được sử dụng trong môi trường có nhiễu nặng, ví dụ khi lắp đặt cáp mạng cùng ống máng với dây cáp điện.

Cáp xoắn cũng được phân chia theo *chất lượng*, được đánh ký hiệu lần lượt là Cat 3, Cat 4, Cat 5, Cat 5e, Cat 6, Cat 6A, Cat 7, Cat 7A, Cat 8, Cat 8.1, Cat 8.2. Các loại cáp này sử dụng trong những công nghệ Ethernet khác nhau, với tốc độ truyền và khoảng cách truyền tối đa cho phép khác nhau.

Trên thị trường hiện nay phổ biến loại cáp Cat 5e và Cat 6. Cáp Cat 5e thường dùng trong công nghệ Fast Ethernet cho phép dữ liệu truyền tối đa 100Mbps với khoảng cách tới 100m. Cáp Cat6 cho phép truyền tối đa tới 10Gbps ở khoảng cách 55m.

Cáp xoắn là loại cáp phổ biến nhất trong mạng LAN hiện nay do giá thành hợp lý, tốc độ truyền dữ liệu cao, dễ thi công. Nhược điểm của loại cáp này là khoảng cách truyền không lớn, có thể bị nhiễu nếu vị trí đi dây hoặc loại dây chọn không phù hợp.

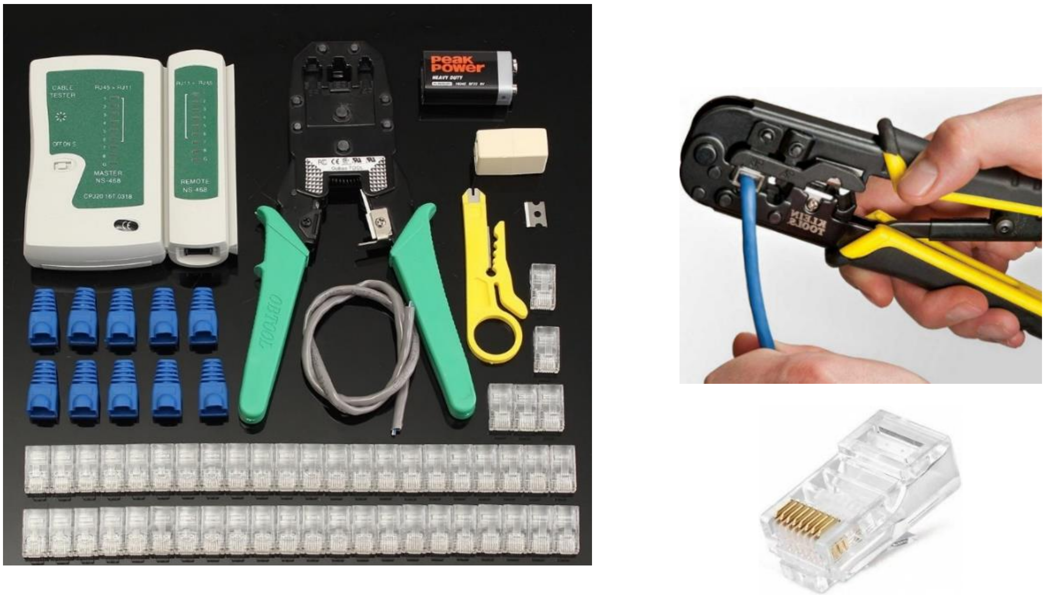
b) Thi công cáp xoắn

Để thi công cáp xoắn, người ta sử dụng bộ dụng cụ thi công bao gồm:

- Đầu kết nối RJ45: để kết nối cáp xoắn với thiết bị mạng hoặc với card mạng của thiết bị đầu cuối.
- Kìm bấm hoặc dao gài chuyên dụng: để ghép đầu nối RJ45 với cáp xoắn.
- Đồng hồ kiểm tra cáp: để kiểm tra chất lượng cáp sau khi bấm.

Đầu kết nối RJ45 là một chuẩn đầu kết nối quốc tế (ISO 8877). Đầu nối RJ45 thường có 8 chân (pin) kim loại. Các dây dẫn của cáp xoắn được ghép vào các chân này.

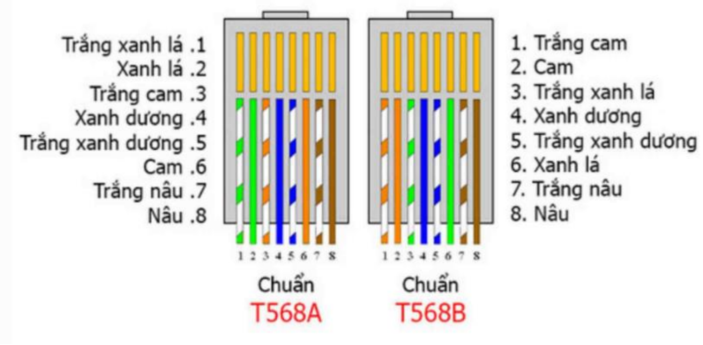
Dây dẫn của cáp xoắn cần ghép với các lá đồng của đầu nối RJ45 theo thứ tự nhất định, gọi là các chuẩn bấm cáp. Có hai chuẩn bấm cáp phổ biến: **T568A** và **T568B** (thường gọi tắt là chuẩn A và chuẩn B - Hình 2.10). Hai chuẩn này khác nhau về thứ tự sắp xếp các dây.



Hình 2.9. Bộ dụng cụ thi công cáp xoắn

Nếu hai đầu của sợi cáp cùng bấm theo chuẩn A hoặc chuẩn B thì sợi cáp đó được gọi là sợi **cáp thẳng**. Cáp thẳng được sử dụng để nối giữa các thiết bị khác loại, ví dụ giữa thiết bị đầu cuối với switch/hub. Hiện nay chuẩn B được sử dụng phổ biến hơn.

Ngược lại một sợi cáp bấm một đầu chuẩn A và một đầu chuẩn B thì sợi cáp đó được gọi là sợi **cáp chéo**. Cáp chéo thường được dùng để kết nối các thiết bị cùng loại, ví dụ giữa thiết bị đầu cuối với thiết bị đầu cuối, giữa router với router.


Trắng xanh lá .1
Xanh lá .2
Trắng cam .3
Xanh dương .4
Trắng xanh dương .5
Cam .6
Trắng nâu .7
Nâu .8
1 2 3 4 5 6 7 8
Chuẩn
T568A
1 2 3 4 5 6 7 8
Chuẩn
T568B
1. Trắng cam
2. Cam
3. Trắng xanh lá
4. Xanh dương
5. Trắng xanh dương
6. Xanh lá
7. Trắng nâu
8. Nâu

Hình 2.10. Thứ tự xếp dây trong 2 chuẩn bấm dây phổ biến

Tuy nhiên, hiện nay hầu hết các cáp nối thiết bị đều là loại cáp thẳng. Các card mạng Gigabit Ethernet cho phép dùng cáp thẳng nối trực tiếp hai thiết bị đầu cuối với nhau. Các switch cũng được nối trực tiếp với nhau bằng cáp thẳng.

Hình 2.11. Một sợi cáp sau khi đã được bấm RJ45 ở hai đầu

Để sử dụng cáp thẳng (straight-through cable) kết nối trực tiếp hai máy tính với nhau, card mạng (Network Interface Card - NIC) trên máy tính cần hỗ trợ tính năng gọi là Auto MDI/MDIX (Automatic Medium-Dependent Interface Crossover). MDI và MDIX là hai kiểu cổng mạng khác nhau: MDI dùng cho thiết bị như máy tính, router; MDIX dùng cho thiết bị trung gian như switch, hub. Nếu không có Auto MDI/MDIX, hai thiết bị cùng loại (như hai máy tính) phải nối bằng cáp bấm chéo để đổi vai trò chân truyền/nhận. Nếu có Auto MDI/MDIX, card mạng sẽ tự động hoán đổi chân truyền và nhận, do đó không cần cáp chéo, chỉ cần cáp thẳng cũng được.

c) Ưu nhược điểm của cáp xoắn

Ưu điểm:

- Giá thành rẻ: Cáp xoắn là loại cáp phổ biến và có chi phí thấp nhất hiện nay trên thị trường, phù hợp với hầu hết các hệ thống mạng quy mô vừa và nhỏ.

- Dễ thi công: Việc lắp đặt cáp xoắn không yêu cầu thiết bị chuyên dụng phức tạp. Kỹ thuật viên chỉ cần các dụng cụ cơ bản như kìm bấm, dao tuốt và đồng hồ kiểm tra cáp.
- Tốc độ truyền cao: Với các loại cáp hiện đại như Cat5e, Cat6, tốc độ truyền dữ liệu có thể đạt từ 100 Mbps đến 10 Gbps tùy theo khoảng cách và chuẩn thiết bị mạng.
- Linh hoạt: Cáp xoắn dễ dàng thi công qua các địa hình khác nhau, có thể đi ngầm, treo trần hoặc qua ống nhựa.
- Tương thích rộng: Hầu hết các thiết bị mạng hiện nay như switch, router, máy tính đều sử dụng cổng RJ45 phù hợp với cáp xoắn.

Nhược điểm:

- Khoảng cách truyền hạn chế: Cáp xoắn chỉ phù hợp cho truyền dữ liệu trong phạm vi khoảng 100 mét. Nếu vượt quá khoảng cách này, cần sử dụng thêm thiết bị khuếch đại hoặc chuyển tiếp tín hiệu.
- Dễ bị nhiễu điện từ: Các loại cáp không có lớp chống nhiễu (như UTP) dễ bị ảnh hưởng bởi môi trường có nhiều sóng điện từ hoặc thiết bị điện công suất lớn.
- Giảm chất lượng theo thời gian: Nếu thi công không đúng kỹ thuật hoặc sử dụng vật tư kém chất lượng, cáp xoắn có thể bị oxi hóa, đứt gãy, hoặc suy hao tín hiệu sau một thời gian sử dụng.
- Hạn chế trong môi trường công nghiệp: Trong môi trường có độ nhiễu cao hoặc yêu cầu bảo mật tín hiệu nghiêm ngặt, cáp xoắn không phải là lựa chọn tối ưu so với cáp quang hoặc cáp có lớp chống nhiễu chuyên dụng.

2.3.3. Cáp quang

Cáp quang là loại cáp truyền dẫn tín hiệu bằng ánh sáng, được sử dụng rộng rãi trong lĩnh vực viễn thông và công nghệ thông tin. Khác với các loại cáp đồng truyền thống truyền tín hiệu điện, cáp quang sử dụng tín hiệu ánh sáng để truyền dữ liệu, giúp tăng tốc độ và độ chính xác trong quá trình truyền thông tin.

Một trong những đặc điểm nổi bật của cáp quang là khả năng truyền dữ liệu với tốc độ rất cao và ổn định, đặc biệt trên những khoảng cách xa mà không bị suy hao đáng kể. Ngoài ra, cáp quang còn có khả năng chống nhiễu tốt do không bị ảnh hưởng bởi các yếu tố điện từ từ môi trường xung quanh. Cáp quang cũng có độ bền cao và thích hợp với môi trường triển khai khắt khe như dưới lòng đất, dưới biển hoặc trong các khu công nghiệp.

Trong thực tế triển khai, các sợi cáp quang thường không được sử dụng đơn lẻ mà được tổ chức thành các bó cáp (hay còn gọi là cáp quang nhiều sợi). Mỗi bó cáp có thể chứa từ vài đến hàng trăm sợi quang, cho phép truyền tải đồng thời nhiều luồng dữ liệu độc lập, đáp ứng nhu cầu kết nối dung lượng lớn trong các mạng viễn thông, trung tâm dữ liệu hoặc hệ thống mạng quy mô lớn.

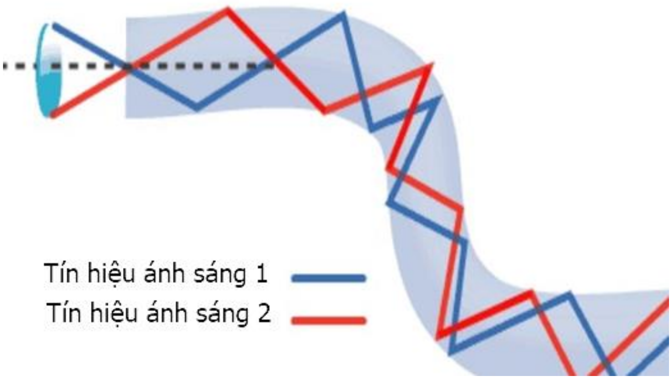
Hình 2.12. Sợi cáp quang gia dụng và bó cáp quang trong công nghiệp

Nhờ những đặc điểm ưu việt, cáp quang được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong mạng Internet tốc độ cao, truyền hình cáp, hệ thống mạng doanh nghiệp, trung tâm dữ liệu, viễn thông quốc tế và các hệ thống giám sát an ninh. Đây là công nghệ nền tảng góp phần xây dựng hạ tầng số hiện đại và phát triển xã hội thông tin.

a) Nguyên lý hoạt động của cáp quang

Nguyên lý truyền tín hiệu trong cáp quang dựa trên hiện tượng **phản xạ toàn phần trong môi trường quang học**. Khi một chùm ánh sáng được truyền vào lõi sợi quang (thường làm từ thủy tinh hoặc nhựa có chiết suất cao), nếu góc tới giữa tia sáng và bề mặt ranh giới giữa lõi và lớp vỏ (cladding) lớn hơn một giá trị nhất định gọi là **góc giới hạn**, ánh sáng sẽ bị phản xạ hoàn toàn trở lại vào bên trong lõi thay vì bị khúc xạ ra ngoài. Quá trình này lặp lại liên tục, giúp tia sáng được dẫn đi theo chiều dài của sợi quang mà không bị thoát ra ngoài.

Tín hiệu ban đầu được tạo ra từ các thiết bị phát như laser hoặc diode phát quang (LED), sau đó được mã hóa dữ liệu và đưa vào sợi quang. Nhờ vào nguyên lý phản xạ toàn phần, tín hiệu ánh sáng có thể được truyền đi xa hàng chục, thậm chí hàng trăm km với mức suy hao rất nhỏ. Ở đầu nhận, bộ thu quang sẽ chuyển đổi tín hiệu ánh sáng ngược trở lại thành tín hiệu điện để xử lý.



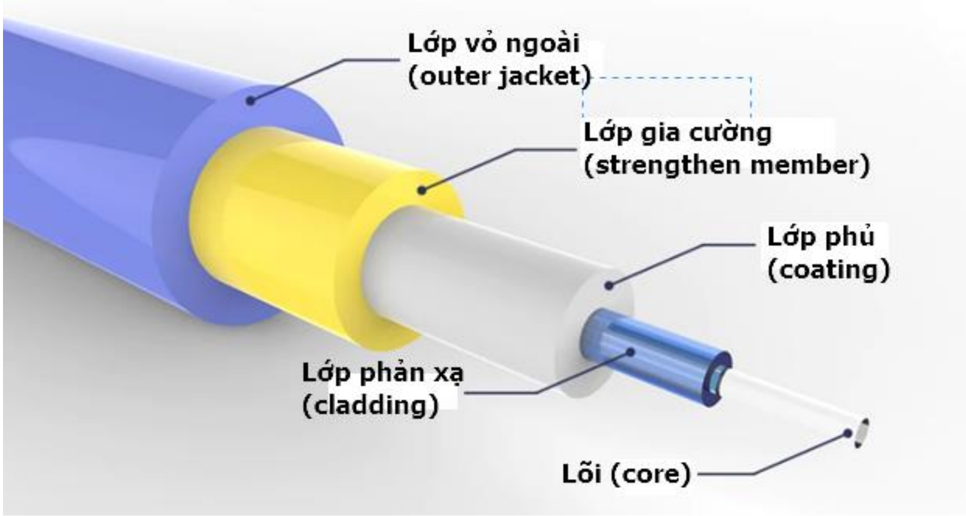
Hình 2.13. Nguyên lý truyền tín hiệu ánh sáng trong cáp quang

Nguyên lý phản xạ toàn phần giúp cáp quang truyền dữ liệu với tốc độ cao, độ chính xác lớn và ít bị nhiễu so với các phương tiện truyền dẫn điện thông thường.

b) Cấu tạo và phân loại cáp quang

Cáp quang bao gồm một hoặc nhiều sợi quang. Mỗi sợi quang gồm các phần sau:

- Sợi lõi (Core): là một sợi thủy tinh hoặc nhựa tinh chế rất nhỏ (kích thước cỡ vài micromet) nằm ở lõi của sợi cáp có tác dụng truyền dẫn ánh sáng.
- Lớp phản xạ (Cladding): là loại vật chất trong suốt có chiết suất thấp bao bọc lõi, có tác dụng tạo ra hiện tượng phản xạ toàn phần khi ánh sáng truyền trong lõi.
- Lớp phủ (Coating): lớp phủ dẻo bằng nhựa bên ngoài có tác dụng bảo vệ sợi khỏi bụi bẩn và hơi nước, giảm gãy gập và uốn cong.
- Lớp gia cường (Strength member): thường làm từ vật liệu Kevlar, có tác dụng chịu nhiệt và chịu kéo căng.
- Lớp vỏ ngoài (Jacket): lớp vỏ bảo vệ bên ngoài chống tác động vật lý và hóa học từ môi trường ngoài.



Hình 2.14. Cấu tạo một sợi cáp quang

Theo cơ chế truyền ánh sáng bên trong lõi, cáp quang được chia làm 2 loại chính là Multimode và Singlemode:

- Cáp quang **Singlemode:** có lõi nhỏ (khoảng 9 micromet), chỉ truyền một bước sóng ánh sáng. Loại cáp này có giá thành cao nhưng truyền tín hiệu đi được rất xa với tốc độ cao.
- Cáp quang **Multimode:** có lõi lớn (gấp 6-8 lần lõi của singlemode), có thể truyền nhiều bước sóng. Nhược điểm của loại cáp này là khoảng cách truyền ngắn, tốc độ chậm. Ưu điểm là giá thành thấp. Cáp multimode có hai cơ chế truyền ánh sáng, gọi là Step Index và Graded Index.

c) Thi công cáp quang

Việc thi công cáp quang khó khăn hơn, yêu cầu các thiết bị chuyên dụng đắt tiền cũng như tay nghề của người thực hiện.

Các công cụ chuyên dụng thường dùng bao gồm:

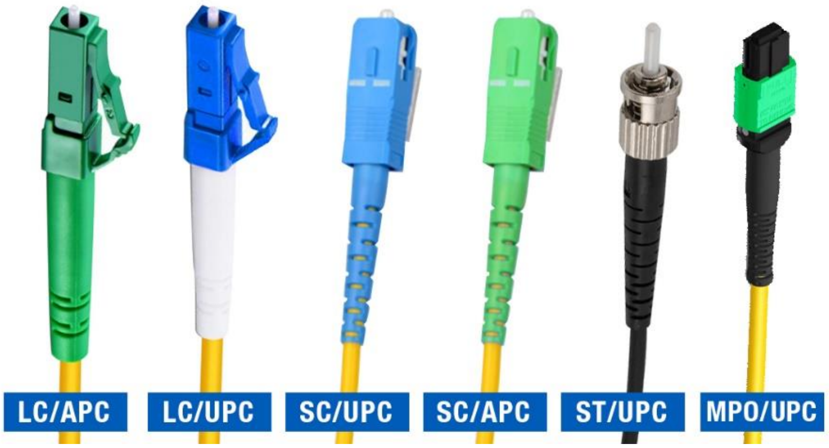
- Dao cắt cáp quang để bóc lớp vỏ bảo vệ bên ngoài,
- Dao tách sợi (fiber stripper) để bóc lớp phủ sợi quang,
- Dao cắt sợi (cleaver) để cắt đầu sợi quang sao cho bề mặt phẳng, nhẵn, phục vụ cho việc hàn nối,
- Máy hàn quang để hàn sợi cáp
- Máy đo công suất quang để kiểm tra suy hao tín hiệu và phát hiện lỗi

Để hàn hai sợi quang lại với nhau, kỹ thuật viên sử dụng máy hàn quang (fusion splicer) - thiết bị này sẽ căn chỉnh tự động và nung chảy hai đầu sợi quang bằng tia hồ quang điện. Sau khi hàn xong, cần dùng lò nung mối hàn để gia cố mối nối bằng ống co nhiệt, giúp bảo vệ sợi quang khỏi gãy gập và môi trường. Ngoài ra, trong quá trình thi công cũng cần đến máy đo công suất quang (optical power meter) và máy soi laser (visual fault locator) để kiểm tra suy hao tín hiệu và phát hiện lỗi trong quá trình đấu nối.

AFL
Optical Power Meter
-5.38
-5.56
OPM5
NOYES
AFL
OLS4
NOYES

Hình 2.15. Máy hàn và đồng hồ đo trong thi công cáp quang

Ngoài các công cụ chuyên dụng, trong thi công cáp quang, kỹ thuật viên cần nắm rõ các loại đầu nối (connector) được sử dụng và phương pháp hàn nối đầu vào sợi quang. Một số loại đầu nối phổ biến hiện nay gồm SC, LC, ST và FC. Các đầu nối này có đặc điểm khác nhau về kích thước, cơ chế khóa (ví dụ SC dùng kiểu cắm-giật nhanh, còn FC dùng kiểu ren vặn chắc chắn), và được lựa chọn tùy theo yêu cầu kỹ thuật của hệ thống.



Hình 2.16. Một số loại đầu nối cáp quang thường gặp

Có hai phương pháp gắn đầu nối vào sợi quang:

- Phương pháp hàn fusion splice và sử dụng pigtail: Kỹ thuật viên sẽ sử dụng một đoạn cáp quang ngắn đã được gắn sẵn đầu connector (gọi là pigtail), sau đó dùng máy hàn quang để hàn nối pigtail với sợi quang chính. Phương pháp này đảm bảo độ suy hao thấp, chất lượng kết nối cao và thường áp dụng cho các hệ thống yêu cầu hiệu suất nghiêm ngặt.
- Phương pháp gắn cơ khí (mechanical splicing): Đây là cách dùng đầu nối có sẵn keo hoặc cơ chế kẹp cơ khí, kỹ thuật viên chỉ cần đưa sợi quang vào đúng vị trí, cố định lại và thực hiện hiệu chỉnh thủ công. Phương pháp này tiện lợi, nhanh chóng nhưng suy hao quang học thường cao hơn so với hàn nhiệt.

d) Ưu nhược điểm của cáp quang

Cáp quang là loại cáp mạng tốt nhất hiện nay, có thể sử dụng cho cả mạng LAN và mạng WAN. Tuy nhiên, cáp quang cũng tồn tại nhiều nhược điểm.

Về ưu điểm, cáp quang có:

- Băng thông lớn và tốc độ cao: Cáp quang hỗ trợ băng thông và tốc độ truyền tải dữ liệu cực cao, vượt trội so với các loại cáp đồng truyền thống.
- Khoảng cách truyền dẫn xa: Cáp quang có khả năng truyền dữ liệu qua khoảng cách rất xa mà không bị suy giảm tín hiệu, lý tưởng cho các kết nối mạng diện rộng (WAN).
- Không bị nhiễu điện từ: Cáp quang không bị ảnh hưởng bởi nhiễu điện từ (EMI) từ các thiết bị điện tử khác, đảm bảo tín hiệu truyền tải ổn định và chính xác.
- Kích thước và trọng lượng nhỏ: Sợi cáp quang có kích thước nhỏ và nhẹ, dễ dàng lắp đặt và quản lý.
- Khả năng bảo mật cao: Cáp quang khó bị nghe lén hoặc xâm nhập trái phép, do đó cung cấp mức độ bảo mật cao hơn cho dữ liệu truyền tải.

Về nhược điểm, cáp quang có:

- Chi phí cao: Chi phí lắp đặt và bảo trì cáp quang thường cao hơn so với cáp đồng, đặc biệt là trong các ứng dụng quy mô lớn.
- Dễ bị hỏng: Sợi cáp quang rất mỏng và dễ bị gãy hoặc hỏng nếu không được xử lý cẩn thận.
- Yêu cầu thiết bị đặc biệt: Cáp quang yêu cầu các thiết bị đặc biệt như bộ chuyển đổi quang điện và các thiết bị đầu cuối quang, làm tăng chi phí và phức tạp trong việc triển khai.
- Khó khăn trong việc sửa chữa: Việc sửa chữa cáp quang đòi hỏi kỹ thuật và thiết bị chuyên dụng, phức tạp hơn so với cáp đồng.

2.3.4. Môi trường truyền không dây

a) Kết nối không dây

Kết nối không dây được thực hiện hơi khác biệt với kết nối sử dụng cáp đồng hay cáp quang. Kết nối không dây thường được thực hiện giữa thiết bị đầu cuối với một thiết bị mạng, gọi là điểm truy cập không dây (Wireless Access Point).

Để thực hiện kết nối không dây, thiết bị đầu cuối phải được lắp đặt một card mạng hỗ trợ kết nối không dây. Card mạng không dây có thể gắn vào khe cắm mở rộng PCI trên mainboard hoặc gắn vào cổng USB. Ngoài ra trên thiết bị đầu cuối cũng thường phải cài đặt phần mềm điều khiển thiết bị (driver) riêng của card mạng không dây đó.

WIRELESS ADAPTER
TP-LINK

Hình 2.17. Card mạng không dây (trái) và điểm truy cập (phải)

Một số loại thiết bị đầu cuối như TV thông minh, điện thoại thông minh, máy tính xách tay đã tích hợp sẵn card mạng không dây. Với máy tính để bàn, card mạng không dây thường phải mua riêng.

Nhiều thiết bị đầu cuối có thể cùng kết nối tới một điểm truy cập không dây. Khi đó điểm truy cập đóng vai trò hạt nhân liên kết các thiết bị đầu cuối thành một mạng máy tính (không dây). Điểm truy cập cũng có thể kết nối với mạng LAN có dây để tạo ra một mạng LAN hỗn hợp (có dây và không dây).

Mạng LAN không dây có thể sử dụng một trong số các công nghệ IEEE 802.11a/b/g/n/ac/ax. Tốc độ truyền dữ liệu tối đa của các chuẩn công nghệ này khá khác biệt. Ví dụ, tốc độ về lý thuyết của chuẩn b là 11Mbps, chuẩn g là 54 Mbps, chuẩn n là 300Mbps. Các thiết bị không dây trong mạng LAN hiện nay chủ yếu sử dụng sóng điện từ dải 2,4GHz (từ 2,401 đến 2,483 GHz) hoặc dải 5GHz (từ 5,470 đến 5,725 GHz).

b) Nguyên lý của truyền tín hiệu không dây

Truyền tín hiệu không dây là quá trình gửi và nhận dữ liệu thông qua sóng điện từ trong không gian, thay vì sử dụng môi trường dẫn truyền vật lý như cáp. Nguyên lý cơ bản của quá trình này dựa trên hiện tượng lan truyền sóng vô tuyến (radio wave propagation) trong không khí.

Quá trình truyền tín hiệu không dây thường bao gồm các bước sau:

- Biến đổi tín hiệu số thành sóng mang điện từ: Dữ liệu số được thiết bị đầu cuối (ví dụ: máy tính, điện thoại) mã hóa và điều chế lên một sóng mang tần số cao. Điều chế là quá trình kết hợp tín hiệu dữ liệu với sóng mang (carrier) để tạo ra tín hiệu có thể truyền đi xa.
- Phát sóng vào môi trường truyền: Tín hiệu đã điều chế được truyền qua anten phát dưới dạng sóng vô tuyến (radio wave), di chuyển trong không gian theo nhiều hướng khác nhau.
- Lan truyền qua môi trường: Trong quá trình lan truyền, sóng có thể bị suy giảm, phản xạ, khúc xạ, hoặc nhiễu loạn bởi các vật cản như tường, kính, kim loại, hoặc các sóng khác trong cùng tần số.
- Thu nhận và giải điều chế: Anten thu của thiết bị nhận bắt được sóng truyền đến, sau đó giải điều chế để tách tín hiệu dữ liệu ra khỏi sóng mang. Tín hiệu thu được sẽ được xử lý và chuyển đổi trở lại thành dữ liệu số.

c) Ưu, nhược điểm của kết nối không dây

Ưu điểm:

- Tính linh hoạt cao: Người dùng có thể truy cập mạng ở bất kỳ đâu trong phạm vi phủ sóng mà không bị giới hạn bởi vị trí cổng kết nối vật lý.
- Dễ dàng triển khai: Việc lắp đặt mạng không dây thường nhanh chóng, không cần kéo dây cáp phức tạp, đặc biệt phù hợp với các không gian khó thi công như tòa nhà cổ, không gian mở hoặc di động.
- Tính di động: Cho phép các thiết bị như laptop, smartphone, máy in không dây di chuyển linh hoạt trong mạng mà vẫn duy trì kết nối.
- Mở rộng dễ dàng: Khi cần thêm thiết bị, chỉ cần cấu hình mà không phải đi lại dây mạng, tiết kiệm chi phí và thời gian mở rộng hệ thống.

Nhược điểm:

- Tốc độ và độ ổn định thấp hơn có dây: Kết nối không dây thường có băng thông thấp hơn và dễ bị gián đoạn do nhiễu sóng, vật cản, hoặc số lượng thiết bị truy cập quá nhiều.

- Bảo mật thấp hơn: Do tín hiệu truyền trong không gian mở, mạng không dây dễ bị tấn công hơn so với mạng có dây nếu không được bảo vệ đúng cách (ví dụ: mã hóa yếu, cấu hình sai).
- Phạm vi hạn chế: Sóng vô tuyến bị suy giảm theo khoảng cách và bị ảnh hưởng bởi vật cản như tường, cửa, thiết bị điện tử…, khiến vùng phủ sóng bị giới hạn.
- Dễ bị nhiễu: Các thiết bị sử dụng cùng tần số như lò vi sóng, điện thoại không dây, hoặc các mạng Wi-Fi lân cận có thể gây nhiễu, làm giảm hiệu năng mạng.

2.4. Thực hành - thi công cáp xoắn

Bài thực hành này hướng dẫn sinh viên cách thi công một sợi cáp xoắn theo chuẩn bấm thẳng (straight-through), thường dùng để kết nối các thiết bị khác loại như máy tính với switch, hoặc switch với router.

1. Chuẩn bị dụng cụ và vật liệu:

- Cáp xoắn loại Cat5e hoặc Cat6 (có chiều dài phù hợp với yêu cầu kết nối).
- Hai đầu nối RJ45 tiêu chuẩn.
- Kìm bấm cáp chuyên dụng.
- Dao tuốt cáp hoặc kéo cắt.
- Đồng hồ kiểm tra cáp (cable tester).
- Bảng màu chuẩn bấm cáp T568B (chuẩn T568B thường dùng hơn).

2. Các bước thi công cáp bấm thẳng:

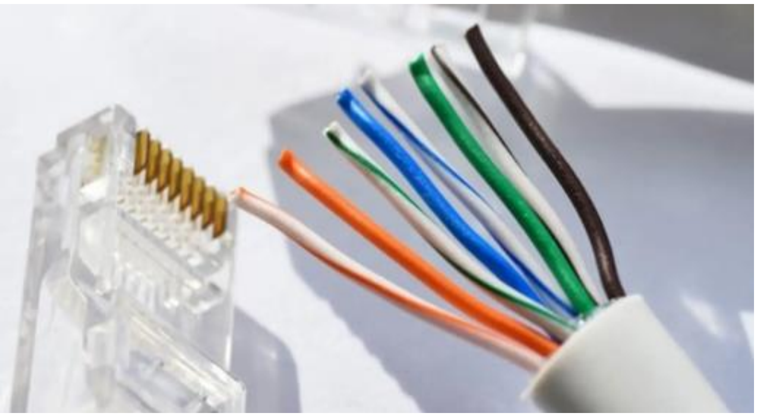
Bước 1: Cắt và tuốt vỏ cáp

- Cắt cáp xoắn theo độ dài cần thiết.
- Dùng dao tuốt hoặc kìm tuốt để loại bỏ khoảng 2-3 cm vỏ ngoài của đầu cáp.
- Gỡ xoắn và tách riêng từng cặp dây dẫn.

Hình 2.18. Cách dùng dao tuốt vỏ cáp

Bước 2: Sắp xếp dây theo chuẩn màu

- Chọn một trong hai chuẩn bấm T568A hoặc T568B (khuyến nghị dùng **T568B**).
- Sắp xếp 8 dây theo đúng thứ tự màu sắc của chuẩn đã chọn.
- Duỗi thẳng các dây và cắt cho đều nhau, chiều dài còn lại khoảng 1.5 cm tính từ mép vỏ cáp.



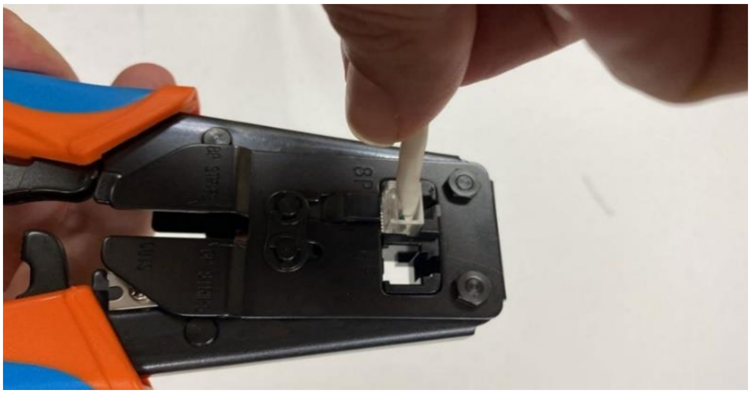
Hình 2.19. Cách xếp dây theo chuẩn T568B

Bước 3: Gắn đầu nối RJ45

- Đẩy các dây đã sắp xếp vào đầu nối RJ45 sao cho đầu dây tiếp xúc hoàn toàn với các chân đồng bên trong.

Hình 2.20. Cách đẩy dây vào đầu nối RJ45

- Kiểm tra lại thứ tự màu sắc trước khi bấm.
- Dùng kìm bấm cáp để bấm chặt đầu RJ45 vào dây.



Hình 2.21. Cách dùng kìm bấm cáp để bấm đầu RJ45

Bước 4: Bấm đầu còn lại

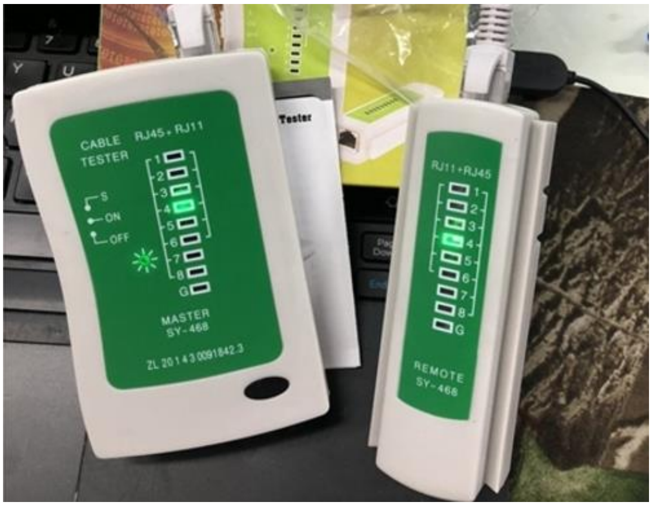
Thực hiện lại các bước từ 1 đến 3 cho đầu cáp còn lại, **sử dụng cùng một chuẩn bấm** với đầu thứ nhất để đảm bảo đó là **cáp thẳng**.

Bước 5: Kiểm tra và hoàn tất

- Kết nối sợi cáp vào thiết bị (ví dụ: máy tính và switch) và kiểm tra tín hiệu hoạt động (đèn kết nối, trạng thái mạng...).

Nếu các thiết bị có hỗ trợ chức năng Auto MDI/MDIX, có thể dùng cáp thẳng để kết nối trực tiếp hai thiết bị cùng loại.

- Dùng sợi cáp vừa bấm kết nối hai máy tính để tạo ra một mạng cơ bản. Sử dụng chương trình ping để kiểm tra kết nối tới máy còn lại.
- Nếu có thể, sử dụng đồng hồ kiểm tra cáp để xác minh các chân đã nối đúng.



Hình 2.22. Sử dụng đồng hồ kiểm tra cáp

Cắm mỗi đầu dây vào một phần của bộ đồng hồ đo và bật đồng hồ. Các chân tương ứng sẽ lần lượt sáng. Đối với cáp thẳng, các chân cùng số phải sáng đồng thời, theo đúng thứ tự từ 1 đến 8.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) Sử dụng kỹ thuật bấm cáp đã học để bấm một dây cáp xoắn theo kiểu cáp chéo. Thử nghiệm kết nối hai máy tính sử dụng cáp chéo và rút ra kết luận.
2) Tìm hiểu cách gài dây cáp xoắn lên hạt cái hoặc patch panel.

Câu hỏi lý thuyết

1) Vì sao tín hiệu số thường được sử dụng trong mạng máy tính thay vì tín hiệu tương tự? Lợi ích nào liên quan đến khả năng chống nhiễu và xử lý số liệu?
2) Giả sử bạn cần truyền tín hiệu mạng qua một khoảng cách 100 mét trong môi trường có nhiễu điện cao (như gần trạm biến áp). Bạn sẽ chọn loại cáp nào? Giải thích lựa chọn dựa trên nguyên lý truyền tín hiệu.
3) Nếu tín hiệu truyền trong cáp bị phản xạ ngược lại (echo), điều gì có thể là nguyên nhân vật lý? Sự cố này ảnh hưởng như thế nào đến dữ liệu?
4) Tại sao ánh sáng lại có thể được dùng để truyền tín hiệu mạng? Nêu ưu điểm và hạn chế của việc sử dụng cáp quang so với cáp đồng xoắn đôi.
5) Tín hiệu truyền trong mạng phải được điều chế theo dạng sóng. Theo bạn, dạng sóng nào (hình sin, hình vuông…) phù hợp nhất cho truyền số? Tại sao?
6) Một sinh viên cho rằng: “Nếu muốn tăng tốc độ mạng, chỉ cần tăng cường độ tín hiệu truyền.” Em có đồng ý không? Giải thích quan hệ giữa băng thông, tốc độ và môi trường truyền.
7) Nếu khoảng cách truyền quá xa mà không sử dụng thiết bị khuếch đại hay lặp tín hiệu, hiện tượng nào sẽ xảy ra với tín hiệu? Giải thích hiện tượng này bằng các kiến thức vật lý mạng.
8) So sánh sự truyền tín hiệu qua môi trường có dây và không dây. Theo bạn, loại nào có độ tin cậy cao hơn trong môi trường doanh nghiệp?
9) Tại sao các chuẩn Ethernet phải quy định giới hạn chiều dài cáp? Giới hạn này liên quan gì đến tính chất lan truyền của tín hiệu?
10) Trong thực tế triển khai mạng tại nhà máy có máy móc lớn và môi trường nhiều sóng điện từ, bạn sẽ đề xuất giải pháp truyền tín hiệu như thế nào để tối ưu hiệu quả và độ ổn định?

BÀI 3. XỬ LÝ DỮ LIỆU TRONG MẠNG MÁY TÍNH

Mục tiêu của bài học

Bài học nhằm giúp sinh viên tiếp cận thực tế quá trình xử lý và truyền dữ liệu trong mạng máy tính thông qua lập trình ứng dụng cơ bản và lý thuyết nền tảng về biểu diễn và đóng/mở gói dữ liệu. Cụ thể, sau khi hoàn thành bài học, sinh viên có thể:

Về kiến thức:

- Nắm được cơ sở biểu diễn dữ liệu trong máy tính thông qua hệ nhị phân, mảng bit, mảng byte.
- Trình bày được khái niệm và ý nghĩa của endian, cách chuyển đổi dữ liệu thành chuỗi byte để phục vụ truyền thông.
- Mô tả quá trình đóng gói và mở gói dữ liệu trong mạng máy tính, từ tầng ứng dụng đến tầng vật lý và ngược lại.

Về kỹ năng:

- Cài đặt và chạy thành công các chương trình truyền thông đơn giản bằng ngôn ngữ C#.
- Chuyển đổi các kiểu dữ liệu cơ bản (số nguyên, số thực, chuỗi) về dạng mảng byte.
- Đọc và phân tích cấu trúc dữ liệu mạng (gói tin) bằng công cụ Wireshark hoặc phần mềm tương tự.

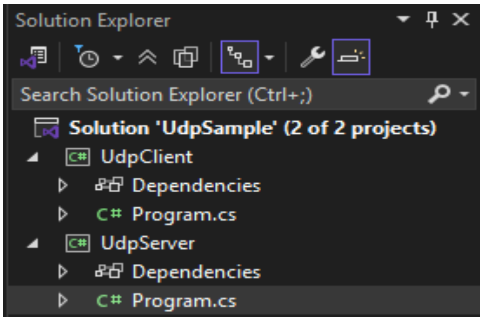
3.1. Thực hành - lập trình truyền thông tin qua mạng cơ bản

Thay vì bắt đầu bằng lý thuyết trừu tượng, chúng ta sẽ tiếp cận thực tế bằng cách trực tiếp lập trình các ứng dụng truyền nhận dữ liệu đơn giản.

Ở phần thực hành này, sinh viên sẽ thực hiện lập trình giao tiếp giữa hai thiết bị sử dụng giao thức UDP. Qua việc xây dựng các ứng dụng client-server cơ bản, sinh viên sẽ trực tiếp trải nghiệm cách thức thiết lập kết nối mạng, gửi - nhận dữ liệu.

Kết quả từ các bài thực hành sẽ tạo nền tảng trực quan vững chắc, giúp sinh viên dễ dàng tiếp cận các lý thuyết về biến đổi thông tin (đóng gói, mở gói) ở các tầng giao tiếp mạng trong phần học tiếp theo.

Bước 1: Tạo hai dự án kiểu Console App, một dự án đặt tên là Client, dự án kia đặt tên là Server. Hai dự án đặt trong giải pháp UdpSocket



Solution 'UdpSample' (2 of 2 projects)
- UdpClient
  - Dependencies
  - Program.cs
- UdpServer
  - Dependencies
  - Program.cs

Hình 3.1. Cấu trúc Solution với hai dự án UdpClient và UdpServer

Bước 2: Thiết lập để đồng thời chạy debug cả hai chương trình.

Bước 3: Mở file Program.cs trong dự án Client và viết code như sau. Chú ý đọc kỹ các comment để nắm được ý nghĩa của từng lệnh.

Client/Program.cs

```
using System.Net.Sockets;
using System.Net;
using System.Text;
using static System.Console;

Title = "UDP Server";
OutputEncoding = Encoding.UTF8;

var localIp = IPAddress.Any;
var localPort = 1308;
var localEndPoint = new IPEndPoint(localIp, localPort);
var socket = new Socket(SocketType.Dgram, ProtocolType.Udp);
socket.Bind(localEndPoint);
WriteLine($"Server hoạt động {localEndPoint}. Chờ dữ liệu tới ...");

while (true) {
    EndPoint remoteEndpoint = new IPEndPoint(IPAddress.Any, 0);
    var receiveBuffer = new byte[16];
    var length = socket.ReceiveFrom(receiveBuffer, ref remoteEndpoint);
    var text = Encoding.ASCII.GetString(receiveBuffer, 0, length);
    PrintArray("Mảng byte nhận được: ", receiveBuffer);
    WriteLine($"Chuỗi ký tự nhận được: {text}");

    receiveBuffer = new byte[4];
    socket.ReceiveFrom(receiveBuffer, ref remoteEndpoint);
    PrintArray("Mảng byte nhận được: ", receiveBuffer);
    var number = BitConverter.ToInt32(receiveBuffer, 0);
    WriteLine($"Số nguyên nhận được: {number}");

    receiveBuffer = new byte[4];
    socket.ReceiveFrom(receiveBuffer, ref remoteEndpoint);
    PrintArray("Mảng byte nhận được: ", receiveBuffer);
    var numberFloat = BitConverter.ToSingle(receiveBuffer, 0);
    WriteLine($"Số thực nhận được: {numberFloat}");
}

void PrintArray(string text, byte[] buffer) {
    WriteLine(text);
    foreach (var b in buffer) {
        Write($"{b} ");
    }
    WriteLine();
}
```

Bước 4: Mở file Program.cs trong dự án Server và viết code như sau:

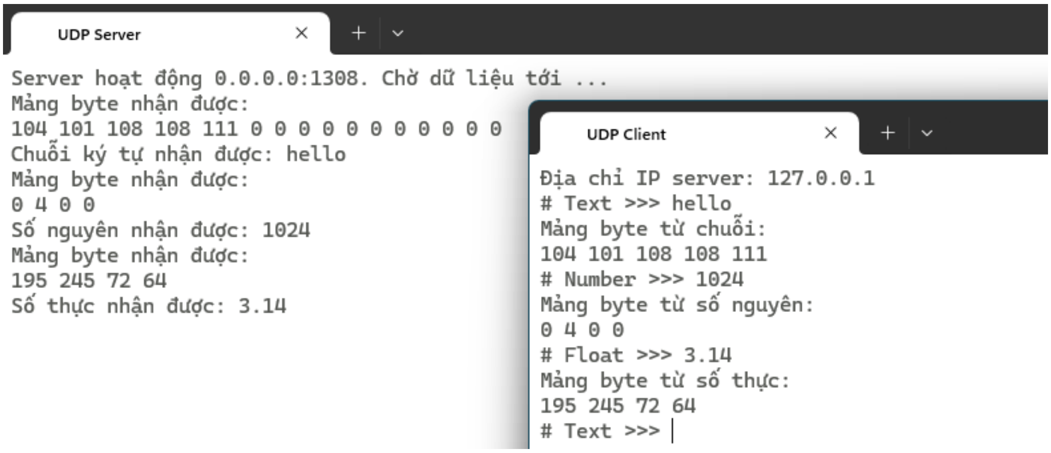
Server/Program.cs

```
using System.Net.Sockets;
using System.Net;
using System.Text;
using static System.Console;

Title = "UDP Client";
OutputEncoding = Encoding.UTF8;

Write("Địa chỉ IP server: ");
var serverIpStr = ReadLine();
var serverIp = IPAddress.Parse(serverIpStr);
var serverPort = 1308;
var serverEndpoint = new IPEndPoint(serverIp, serverPort);
while (true) {
    Write("# Text >>> ");
    ResetColor();
    var text = ReadLine();
    var socket = new Socket(SocketType.Dgram, ProtocolType.Udp);
    var sendBuffer = Encoding.ASCII.GetBytes(text);
    PrintArray("Mảng byte từ chuỗi: ", sendBuffer);
    socket.SendTo(sendBuffer, serverEndpoint);

    Write("# Number >>> ");
    var number = int.Parse(ReadLine());
    var numberBuffer = BitConverter.GetBytes(number);
    PrintArray("Mảng byte từ số nguyên: ", numberBuffer);
    socket.SendTo(numberBuffer, serverEndpoint);

    Write("# Float >>> ");
    var numberFloat = float.Parse(ReadLine());
    var numberFloatBuffer = BitConverter.GetBytes(numberFloat);
    PrintArray("Mảng byte từ số thực: ", numberFloatBuffer);
    socket.SendTo(numberFloatBuffer, serverEndpoint);
```

```
   socket.Close();
}

void PrintArray(string text, byte[] buffer) {
   WriteLine(text);
   foreach (var b in buffer) {
      Write($"{b} ");
   }
   WriteLine();
}
```

Bước 5: Chạy thử ứng dụng bằng phím F5 hoặc tổ hợp Ctrl+F5. Khi này, chương trình server sẽ chạy trước, chương trình client sẽ chạy ngay sau server.

Chương trình client yêu cầu người dùng lần lượt nhập vào một chuỗi ký tự, một số nguyên, và một số thực. Chương trình client sẽ biến đối các dữ liệu này thành chuỗi byte và truyền tới chương trình server. Cách thức biến đổi dữ liệu thành chuỗi byte cũng như quá trình thực tế diễn ra trong quá trình biến đổi và truyền thông tin sẽ được xem xét chi tiết ở phần tiếp theo của bài học này.



```
Server hoạt động 0.0.0.0:1308. Chờ dữ liệu tới ...
Mảng byte nhận được:
104 101 108 108 111 0 0 0 0 0 0 0 0 0 0 0
Chuỗi ký tự nhận được: hello
Mảng byte nhận được:
0 4 0 0
Số nguyên nhận được: 1024
Mảng byte nhận được:
195 245 72 64
Số thực nhận được: 3.14
```

```
Địa chỉ IP server: 127.0.0.1
# Text >>> hello
Mảng byte từ chuỗi:
104 101 108 108 111
# Number >>> 1024
Mảng byte từ số nguyên:
0 4 0 0
# Float >>> 3.14
Mảng byte từ số thực:
195 245 72 64
# Text >>>
```

Hình 3.2. Kết quả hoạt động của chương trình truyền thông tin qua UDP

Lưu ý, khi Server hoạt động lần điều tiên, hệ điều hành sẽ yêu cầu người dùng cấp phép cho Server chiếm dụng cổng 1308. Nếu chạy Server từ một tài khoản bị hạn chế quyền sẽ không thực hiện được yêu cầu này, và Server sẽ bị đóng lại ngay sau đó. Từ lần hoạt động sau sẽ không cần trả lời cấp phép nữa.

3.2. Biến đổi dữ liệu trong mạng máy tính

Trong mạng máy tính, dữ liệu người dùng không trực tiếp truyền đi. Mọi dữ liệu người dùng đều cần được chuyển đổi về một chuỗi byte lớn trước khi biến đổi thành tín hiệu có thể truyền qua môi trường vật lý. Ở thiết bị đích diễn ra quá trình đảo ngược, trong đó tín hiệu biến trở lại chuỗi byte lớn, sau đó chuyển đổi trở lại thành dữ liệu người dùng. Quá trình chuyển đổi dữ liệu người dùng thành chuỗi byte trên máy nguồn gọi là *đóng gói dữ liệu*. Quá trình chuyển đổi chuỗi byte thành dữ liệu người dùng trên máy đích gọi là *mở gói dữ liệu*.

Trước khi tìm hiểu chi tiết cách đóng gói và truyền dữ liệu, chúng ta cần nắm vững cơ sở biểu diễn dữ liệu trong máy tính và mạng máy tính, đó là hệ nhị phân, mảng bit, mảng byte.

3.2.1. Hệ nhị phân, chuỗi bit, mảng byte

Hệ nhị phân (binary system) là hệ đếm cơ bản nhất trong máy tính và mạng máy tính, sử dụng chỉ hai ký tự **0** và **1** để biểu diễn tất cả các giá trị. Mọi dữ liệu, từ ký tự, số nguyên, số thực cho đến hình ảnh, video đều được mã hóa dưới dạng các chuỗi bit nhị phân. Máy tính chỉ hiểu và xử lý thông tin dựa trên các dãy số 0 và 1 này.

Ví dụ:

- Số thập phân 5 được biểu diễn trong hệ nhị phân là 101.
- Chữ cái A trong bảng mã ASCII có mã nhị phân là 01000001.

Mảng bit (bit array), hoặc **chuỗi bit**, là một tập hợp các bit được lưu trữ liên tiếp nhau. Trong mạng máy tính, dữ liệu cần truyền đi sẽ được tổ chức thành các mảng bit trước khi được đóng gói thành gói tin để biến thành tín hiệu vật lý.

Mặc dù về mặt toán học, một số biểu diễn ở hệ nhị phân có thể có số bit biến động. Ví dụ, số 5 (thập phân) được biểu diễn với 3 bit (101), số 9 (thập phân) được biểu diễn với 4 bit (1001). Tuy nhiên, để xử lý trong máy tính, người ta phải sử dụng số lượng bit cố định cho từng kiểu dữ liệu. Ví dụ: số 9 (thập phân) có thể được biểu diễn dưới dạng mảng 8 bit 00001001. Trong 8 bit này, thực tế chỉ cần 4 bit cho số 9, 4 bit còn lại để 0. Cùng số 9 (thập phân) cũng có thể được biểu diễn ở dạng mảng 16 bit 00000000 00001001.

Việc lựa chọn số lượng bit để biểu diễn giá trị tùy thuộc vào dải giá trị mong muốn của kiểu dữ liệu. Thông thường, người ta sử dụng 8 bit / 16 bit / 32 bit / 64 bit cho các kiểu dữ liệu. Ví dụ, để biểu diễn số nguyên, người ta có thể sử dụng 8 bit (kiểu byte), 16 bit (kiểu dword), 32 bit (kiểu qword). Để biểu diễn số thực, người ta thường sử dụng 32 bit (kiểu float) hoặc 64 bit (kiểu double). Để biểu diễn giá trị logic (true / false), người ta thường dùng 8 bit. Với kiểu ký tự, người ta có thể sử dụng 8 bit (ASCII), 8-16-32 bit (Unicode UTF-8).

Tùy thuộc vào số bit sử dụng để biểu diễn giá trị, mỗi kiểu dữ liệu sẽ có số lượng và dải giá trị xác định. Sử dụng n bit cho phép biểu diễn tối đa 2^n giá trị. Ví dụ, dùng 8 bit để biểu diễn số nguyên thì có thể biểu diễn tới 2^8 (256) giá trị, với dải giá trị có thể tùy thuộc thiết kế của kiểu (-127 - 127, hoặc 0 - 255).

Mặc dù đơn vị dữ liệu trong máy tính là bit, việc xử lý bit chỉ được thực hiện ở cấp độ thấp. Ở cấp độ ứng dụng, các bit dữ liệu được nhóm lại để tăng hiệu quả xử lý. Một nhóm 8 bit liên tục tạo thành một **byte**. Một nhóm nhiều byte nối tiếp nhau tạo thành **mảng byte** (hoặc chuỗi byte).

Kiểu mảng byte trong các ngôn ngữ lập trình tương tự C là byte[]

Mảng byte được sử dụng khi truyền dữ liệu người dùng qua mạng. Tất cả dữ liệu người dùng đều được biến đổi thành mảng byte trước khi cung cấp cho dịch vụ truyền thông mạng. Mỗi kiểu dữ liệu cơ sở chuyển về dạng mảng byte theo cách khác nhau, tùy thuộc vào kích thước và đặc thù của loại dữ liệu đó. Ví dụ, 1 số nguyên sử dụng 4 byte để biểu diễn nên khi chuyển về dạng mảng byte, nó sẽ bao gồm 4 byte viết kế tiếp nhau.

3.2.2. Biểu diễn chuỗi bit / byte

a) Biểu diễn một chuỗi bit / byte cụ thể

Một mảng byte cụ thể có thể được biểu diễn bằng một trong những cách sau: sử dụng giá trị ở hệ thập phân, sử dụng giá trị ở hệ thập lục phân, sử dụng giá trị ở hệ nhị phân.

Dạng thập lục phân (hệ cơ số 16) là cách phổ biến nhất khi mô tả mảng byte trong mạng hoặc lập trình hệ thống. Ví dụ: `0x 0A 1F 2B 00 FF 3C` là một chuỗi byte được biểu diễn ở hệ cơ số 16. Mỗi cặp giá trị của chuỗi là một byte (8 bit), viết theo hệ 16. Trong lối viết này, người ta thường thêm tiền tố 0x vào trước. Cách viết này dễ đọc, ngắn gọn, phù hợp với phân tích gói tin. Lối viết này thường gặp trong biểu diễn địa chỉ MAC hoặc địa chỉ IPv6.

Dạng giá trị nhị phân (bit level) dùng để thể hiện chi tiết từng bit trong từng byte, phù hợp với biểu diễn header hoặc giao thức. Ví dụ: `00001010 00011111 00101011 00000000 11111111 00111100`. Mỗi nhóm 8 ký tự là một byte. Lối viết này thường dùng trong phân tích và biến đổi địa chỉ IPv4.

Dạng số thập phân là dạng biểu diễn dễ hiểu với người không chuyên, nhưng dài và khó nhận diện giá trị ở mức độ bit. Ví dụ `10 31 43 0 255 60` là một chuỗi byte biểu diễn ở dạng thập phân. Lối viết này thường gặp trong biểu diễn địa chỉ IPv4.

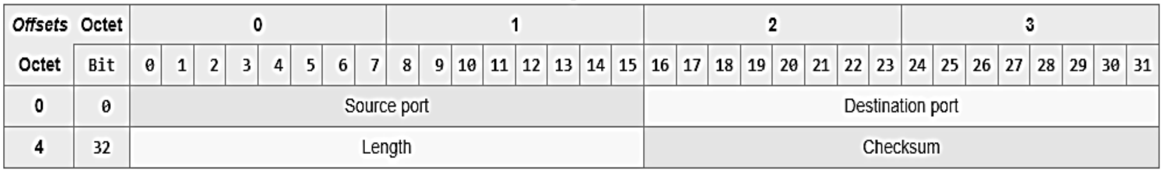
Do một chuỗi bit thường rất dài, chúng ta không thể viết nó trên cùng một dòng. Ngoài ra, việc viết cả chuỗi bit / byte thường chỉ phù hợp để biểu diễn một chuỗi bit / byte cụ thể. Lối viết chuỗi bit / byte như vậy không phù hợp để mô tả cấu trúc của chuỗi bit / byte. Do đó, người ta sử dụng một cách viết khác trực quan hơn để mô tả cấu trúc chung của chuỗi bit / byte.

b) Mô tả cấu trúc chuỗi bit / byte

Để mô tả cấu trúc của chuỗi bit / byte, người ta sử dụng cách viết gần giống như biểu diễn một ma trận. Cụ thể, người ta thường chỉ viết 32 bit (4 byte) trên mỗi dòng, và nhóm mỗi 8 bit vào một octet/byte. Mỗi nhóm byte / bit liên tục kết hợp với nhau để diễn đạt một ý nghĩa và có dải giá trị xác định được gọi là một **trường** (field). Mỗi trường có tên gọi riêng để phân biệt. Một chuỗi bit/byte khi đó là một tập hợp các trường.

Để tiện lợi cho việc tính toán ra vị trí của bit hoặc byte trong mảng, người ta đưa vào hai dòng phụ: dòng đầu đánh số byte từ 0 đến 3 (dòng Octet), dòng 2 đánh số bit từ 0 đến 31 (dòng Bit). Đầu mỗi dòng người ta thêm giá trị offset của byte và offset của bit.

Vị trí của byte/octet ở mỗi dòng tính bằng offset của byte/octet ở dòng đó cộng với số thứ tự cột tương ứng ở dòng Octet. Tương tự, vị trí của bit ở mỗi dòng bằng offset của bit của dòng đó cộng số thứ tự của cột tương ứng ở dòng Bit.



Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Hình 3.3. Ví dụ về cách biểu diễn một chuỗi byte

Đây là một ví dụ sử dụng hai dòng để mô tả một mảng 8 byte (64 bit), mỗi dòng chứa 4 byte (32 bit). Mảng này được chia làm 4 phần:

- *Source port* chiếm từ bit 0 đến bit 15 (tức là byte 0 và byte 1),
- *Destination port* chiếm từ bit 16 đến 31 (tức là byte 2 và byte 3),
- *Length* chiếm từ bit 32 + 0 = 32 (32 là bit offset của dòng 2) đến 32 + 15 = 47 (tức là byte 4 + 0 = 4 và byte 4 + 1 = 5, với 4 là byte offset của dòng 2),
- *Checksum* chiếm từ bit 32 + 16 = 48 tới bit 32 + 31 = 63 (tức là byte 4 + 2 = 6 và byte 4 + 3 = 7).

Cách viết này giúp chúng ta mô tả cấu trúc chi tiết của một chuỗi byte/bit có độ dài bất kỳ trên mặt giấy một cách tiện lợi.

3.2.3. Khái niệm endian

Khi biểu diễn một số ở dạng mảng byte chúng ta sẽ gặp một vấn đề: trình tự thể hiện các byte liên quan trong một chuỗi. Ví dụ, số nguyên 1024 biểu diễn ở dạng hexa là `0x0400`. Số nguyên trong các ngôn ngữ lập trình thường có độ dài là 4 byte, trong khi `0x0400` hiện chỉ có độ dài 2 byte. Do đó phải bổ sung thêm 2 byte `00 00` vào cho đủ 4 byte. Ở đây phát sinh một vấn đề: có 2 cách thức khác nhau để viết 4 byte này: `0x00 00 04 00` và `0x 00 04 00 00`. Hai cách thức này khác nhau ở thứ tự sắp xếp các byte trong một chuỗi.

Thứ tự viết các byte (với các kiểu dữ liệu chiếm nhiều byte) được gọi là **endianness**. Endian là một thuật ngữ dùng để chỉ thứ tự byte được sử dụng để biểu diễn một giá trị dữ liệu đa byte trong bộ nhớ máy tính hoặc khi truyền dữ liệu qua mạng. Khi một giá trị lớn hơn 1 byte (ví dụ: 2 byte, 4 byte, 8 byte) được lưu trữ hoặc truyền đi, có hai cách phổ biến để sắp xếp thứ tự các byte tạo thành giá trị đó, gọi là Little-Endian và Big-Endian.

- Big-Endian:
  - Byte có ý nghĩa lớn nhất (Most Significant Byte - MSB) được lưu ở địa chỉ thấp nhất, tức là byte đầu tiên trong dãy dữ liệu là phần quan trọng nhất.
  - Ví dụ: Số `0x12345678` sẽ được lưu trong bộ nhớ (theo thứ tự địa chỉ tăng dần) là 0x`12 34 56 78`.
- Little-Endian:
  - Byte có ý nghĩa nhỏ nhất (least significant byte - LSB) được lưu ở địa chỉ thấp nhất, tức là byte đầu tiên là phần ít quan trọng nhất.
  - Ví dụ: Số `0x12345678` sẽ được lưu là `78 56 34 12`.

Big-Endian tương tự như cách chúng ta viết số từ trái sang phải. Chữ số bên phải giá trị kém 256 lần chữ số bên trái. Little-Endian tương tự như cách chúng ta viết và đọc số từ phải qua trái, nghĩa là chữ số bên tay phải có giá trị cao hơn 256 lần chữ số bên tay trái.

Trong lĩnh vực mạng, thứ tự byte là rất quan trọng vì các máy tính có thể sử dụng các chuẩn endian khác nhau. Hệ điều hành Windows sử dụng lối viết Little-Endian. Hệ điều hành gốc Linux và Unix sử dụng lối viết Big-Endian.

Để đảm bảo tính nhất quán khi truyền dữ liệu qua mạng, các giao thức như TCP/IP quy định sử dụng Big-Endian, còn được gọi là *trật tự byte mạng* (network byte order). Khi máy tính gửi dữ liệu lên mạng, nếu nó sử dụng Little-Endian (như máy tính Intel), nó phải chuyển đổi dữ liệu sang Big-Endian trước khi truyền đi, và ngược lại khi nhận về.

Lưu ý, khái niệm Endian chỉ áp dụng với chuỗi byte. Với chuỗi bit không có khái niệm Endian. Một chuỗi bit luôn viết theo lối viết số thông thường chúng ta sử dụng (tức là tương tự như Big-Endian).

3.2.4. Biến đổi dữ liệu về mảng byte

a) Chuyển đổi số nguyên về mảng byte

Số nguyên (kiểu `int`, `short`, `long`, ...) trong máy tính được biểu diễn bằng một số cố định các byte (thường là 1, 2, 4 hoặc 8 byte). Quá trình chuyển đổi số nguyên thành mảng byte tuân theo các bước sau:

- Biểu diễn số nguyên dưới dạng nhị phân có độ dài phù hợp với số byte của kiểu dữ liệu.
- Cắt chuỗi bit nhị phân thành các nhóm 8 bit, mỗi nhóm tương ứng một byte.
- Sắp xếp các byte theo thứ tự byte của hệ thống:
  - **Little-endian**: byte có trọng số thấp đặt trước (ví dụ: Intel, Windows).
  - **Big-endian**: byte có trọng số cao đặt trước (ví dụ: mạng TCP/IP).

Ví dụ: Số nguyên `1024` (thập phân) có giá trị nhị phân `00000100 00000000`. Nếu dùng 4 byte và biểu diễn theo Little-Endian, mảng byte là `0x 00 04 00 00`. Nếu biểu diễn theo dạng Big-Endian chúng ta thu được mảng `0x 00 00 04 00`.

Trong ngôn ngữ C#, việc chuyển đổi có thể thực hiện như sau:

```
int value = 1024;
byte[] bytes = BitConverter.GetBytes(value); // Mặc định là Little-endian
```

b) Chuyển đổi chuỗi ký tự về mảng byte

Chuỗi ký tự là dạng dữ liệu phổ biến nhất, bao gồm các ký tự chữ cái, số, ký hiệu,... Mỗi ký tự cần được ánh xạ sang một hoặc nhiều byte tùy theo **bảng mã hóa** sử dụng.

Một số bảng mã phổ biến:

- **ASCII**: mỗi ký tự chiếm đúng 1 byte (8 bit), chỉ hỗ trợ ký tự Latin.
- **UTF-8**: mỗi ký tự chiếm 1-4 byte, hỗ trợ toàn bộ Unicode.
- **Unicode/UTF-16**: mỗi ký tự chiếm 2 hoặc 4 byte.

Quy trình chuyển đổi:

- Lấy từng ký tự trong chuỗi.
- Mã hóa sang giá trị số nguyên tương ứng theo bảng mã.
- Gộp các giá trị thành mảng byte theo thứ tự gốc.

Ví dụ: Chuỗi `"HELLO"` sử dụng mã ASCII được biểu diễn thành mảng byte `0x 48 45 4C 4C 4F`.

> Bảng mã ASCII và UTF-8 không chịu ảnh hưởng của endian. Bảng mã ASCII sử dụng 1 byte cho mỗi ký tự. Bảng mã UTF-8 mặc dù có thể sử dụng nhiều byte cho một ký tự nhưng các byte có liên kết với nhau theo quy tắc xác định.

Trong ngôn ngữ C# có thể thực hiện chuyển đổi chuỗi ký tự về mảng byte như sau:

```
string text = "HELLO";

byte[] bytes = Encoding.ASCII.GetBytes(text);
string text = "Xin chào";
byte[] bytes = Encoding.UTF8.GetBytes(text); // UTF-8 hỗ trợ tiếng Việt
```

c) Chuyển đổi số thực về mảng byte

Số thực (kiểu `float` hoặc `double`) được biểu diễn ở dạng nhị phân theo chuẩn **IEEE 754**. Kiểu `float` chiếm 4 byte (32 bit), `double` chiếm 8 byte (64 bit), với cấu trúc chung bao gồm gồm: 1 bit dấu, một trường số mũ, và một phần trị tuyệt đối (mantissa).

Chuẩn **IEEE 754** là tiêu chuẩn quốc tế định nghĩa cách **biểu diễn số thực** trong máy tính nhị phân. Chuẩn này được sử dụng rộng rãi trong phần lớn các hệ thống máy tính, ngôn ngữ lập trình và giao thức mạng.

Dưới đây là mô tả chi tiết cho hai kiểu chính:

Kiểu float

Thành phần	Số bit	Vị trí
Bit dấu (Sign)	1	Bit 31 (MSB)
Số mũ (Exponent)	8	Bit 30 → 23
Phần trị (Mantissa)	23	Bit 22 → 0

$$\text{Giá trị} = (-1)^{\text{sign}} \times 1.\text{fraction} \times 2^{\text{exponent}-127}$$

Kiểu double

Thành phần	Số bit	Vị trí
Bit dấu (Sign)	1	Bit 63
Số mũ (Exponent)	11	Bit 62 → 52
Phần trị (Mantissa)	52	Bit 51 → 0

$$\text{Giá trị} = (-1)^{\text{sign}} \times 1.\text{fraction} \times 2^{\text{exponent}-1023}$$

Quá trình chuyển đổi số thực thành mảng byte gồm:

- Mã hóa giá trị số thực theo chuẩn IEEE 754.
- Chia thành các byte theo thứ tự endianness.

Ví dụ: Số thực `3.14f` có biểu diễn IEEE 754 dạng hexa là `0x 40 48 F5 C3` (Big-endian) hoặc `0x C3 F5 48 40` (Little-endian).

Trong ngôn ngữ C#, việc chuyển đổi số thực về mảng byte như sau:

```
float value = 3.14f;
byte[] bytes = BitConverter.GetBytes(value); // Little-endian
```

3.2.5. Đóng mở gói dữ liệu

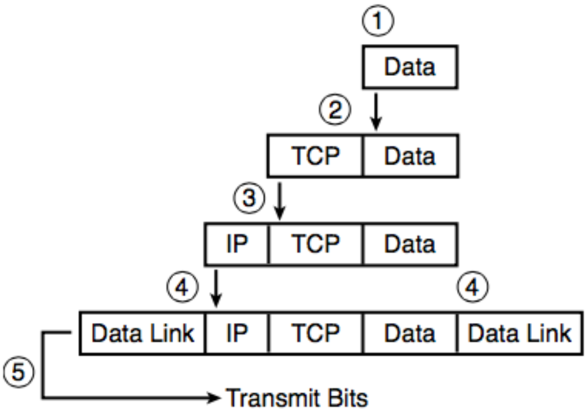
Sau khi nắm được cách biến đổi dữ liệu ở dạng chuỗi byte, chúng ta sẽ xem xét cách biến đổi chuỗi byte thực tế diễn ra trong hệ thống mạng khi trao đổi thông tin.

Trên *thiết bị nguồn*, dữ liệu trải qua một quá trình phức tạp trước khi biến thành tín hiệu truyền đi trên môi trường truyền dẫn mạng. Quá trình này gọi là **đóng gói dữ liệu**.

Có thể tóm tắt quá trình đóng gói như sau:

1. Chương trình ứng dụng chuẩn bị dữ liệu ở dạng chuỗi byte.
2. Chương trình chuyển chuỗi byte dữ liệu cho chương trình TCP hoặc UDP. Chương trình bắt buộc phải lựa chọn sử dụng chương trình TCP hoặc UDP, tùy vào yêu cầu của việc truyền thông tin.
3. Chương trình TCP hoặc UDP sẽ tạo ra một chuỗi byte mới dựa trên dữ liệu của ứng dụng và chuyển tiếp cho chương trình IP.

4. Chương trình IP tạo một chuỗi byte mới gọi là packet dựa trên dữ liệu do TCP hoặc UDP chuyển xuống và chuyển xuống cho card mạng.
5. Card mạng tạo một chuỗi byte mới gọi là frame (dựa trên dữ liệu của IP đưa xuống) và chuyển thành tín hiệu đưa lên đường truyền.



Hình 3.4. Trình tự đóng gói chuỗi byte ở thiết bị nguồn với TCP

Sau đây là chi tiết các bước.

a) Chuyển dữ liệu người dùng về chuỗi byte

Dữ liệu người dùng là dữ liệu do người dùng tạo ra hoặc dữ liệu phục vụ cho người dùng thông qua phần mềm. Dữ liệu người dùng tồn tại ở nhiều dạng khác nhau, như văn bản, số, logic (boolean), object, video, audio, ảnh.

Dữ liệu người dùng cần được phần mềm biến đổi về dạng chuỗi byte. Đây là yêu cầu bắt buộc vì tất cả các chương trình trong mạng đều chỉ chấp nhận xử lý các chuỗi byte. Cách biến đổi dữ liệu về chuỗi byte phụ thuộc vào từng loại dữ liệu người dùng cụ thể mà chúng ta đã xem xét ở phần trên.

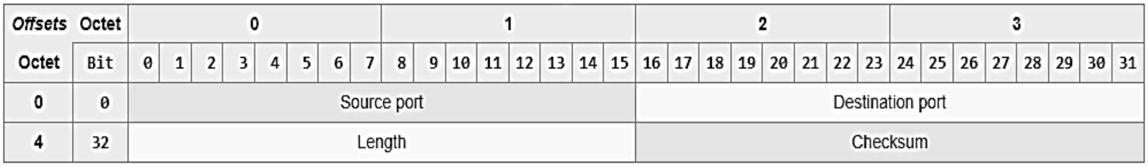
Từ đây về sau, tất cả những gì mà các chương trình mạng xử lý đều là những mảng byte. Dữ liệu người dùng chỉ tồn tại đối với chương trình ứng dụng.

b) Đóng gói dữ liệu với TCP hoặc UDP

UDP và TCP là hai chương trình con do hệ điều hành quản lý. Hai chương trình con này cung cấp một **giao diện lập trình ứng dụng** (API - Application Programming Interface, gọi tắt là **socket API**), cho phép chương trình khác gọi đến, tương tự như việc gọi hàm trong lập trình.

Chương trình ứng dụng bắt buộc phải lựa chọn gọi một trong hai chương trình, TCP hoặc UDP, tùy vào yêu cầu của việc truyền dữ liệu. Chuỗi byte dữ liệu được chương trình chuyển xuống cho chương trình TCP/UDP theo cơ chế tương tự như việc gọi hàm và truyền tham số.

Chương trình UDP và TCP sẽ bổ sung một chuỗi byte nhỏ vào đầu chuỗi byte dữ liệu nó nhận được để tạo ra một chuỗi byte lớn hơn. Chuỗi byte bổ sung này tên gọi là header hoặc **PCI** (Protocol Control Information). Header của UDP và TCP có cấu trúc xác định riêng.



<table>
<tr><td>Offsets</td><td>Octet</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Octet</td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td>0</td><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><td>4</td><td>32</td><td colspan="16">Length</td><td colspan="16">Checksum</td></tr>
</table>

Hình 3.5. Cấu trúc chuỗi byte header của UDP

Chuỗi byte lớn do TCP tạo ra gọi là **segment**. Chuỗi byte do UDP tạo ra gọi là **datagram**. Cả TCP segment và UDP datagram được gọi chung là **PDU** (Protocol Data Unit).

Chi tiết về cấu trúc của các chuỗi byte header của TCP và UDP sẽ được xem xét ở bài học tiếp theo.

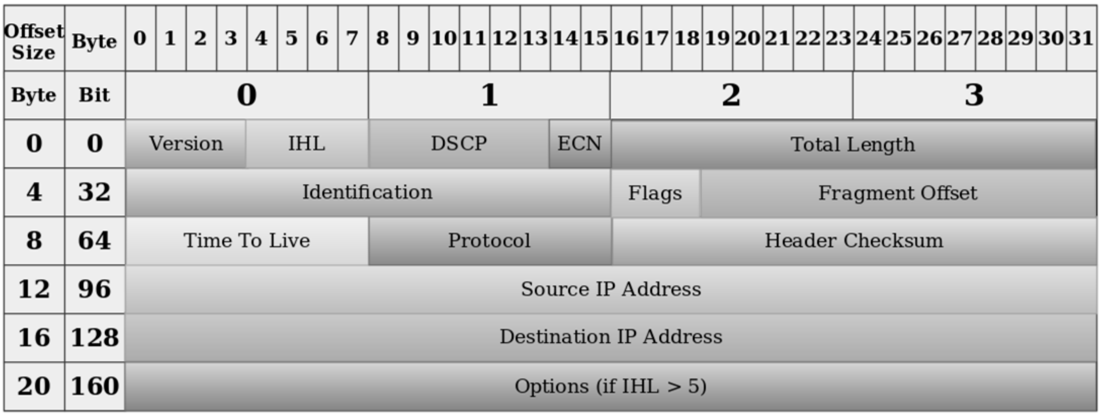
Tiếp theo, TCP (hoặc UDP) tự động gọi và chuyển chuỗi byte segment (hoặc datagram) cho chương trình IP.

c) Đóng gói dữ liệu với IP

IP cũng là một chương trình con do hệ điều hành quản lý (tương tự như TCP và UDP). IP cũng cung cấp socket API cho phép chương trình khác gọi nó theo cách thức tương tự như gọi hàm. IP socket API được gọi tự động bởi chương trình TCP hoặc UDP.

Phần mềm ứng dụng có thể (và bắt buộc) phải gọi trực tiếp tới IP socket API để xây dựng một số chương trình đặc biệt như ping, tracert. Khi này phần mềm bỏ qua việc gọi TCP hoặc UDP socket.

IP chế tạo PDU bằng cách đặt thêm chuỗi byte header vào đầu chuỗi TCP segment (hoặc UDP dgram) vừa nhận được. Chuỗi byte lớn do IP tạo ra này được gọi là packet.



<table>
<tr><td>Offset Size</td><td>Byte</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>Byte</td><td>Bit</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>0</td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><td>4</td><td>32</td><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td>8</td><td>64</td><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td>12</td><td>96</td><td colspan="32">Source IP Address</td></tr>
<tr><td>16</td><td>128</td><td colspan="32">Destination IP Address</td></tr>
<tr><td>20</td><td>160</td><td colspan="32">Options (if IHL > 5)</td></tr>
</table>

Hình 3.6. Cấu trúc chuỗi byte header của IP

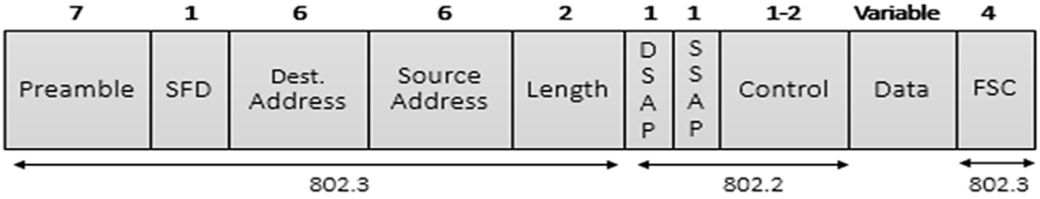
IP chuyển tiếp packet này tới card mạng. Sau khi đóng gói packet và chuyển đến card mạng, hệ điều hành không còn kiểm soát quá trình đóng gói dữ liệu nữa.

d) Đóng gói dữ liệu phục vụ truyền thông trong LAN

Card mạng, còn gọi là Card giao diện mạng (NIC - Network Interface Card) là một thiết bị phần cứng được gắn vào máy tính hoặc thiết bị đầu cuối, cho phép thiết bị đó kết nối và giao tiếp với môi trường truyền dẫn của mạng máy tính. NIC cho phép chạy chương trình ở dạng *firmware*. Tùy thuộc vào môi trường truyền dẫn sẽ cần đến loại NIC phù hợp, với firmware riêng.

Firmware là một loại phần mềm đặc biệt được nhúng sẵn trong phần cứng, dùng để điều khiển và quản lý hoạt động của phần cứng đó. Không giống như phần mềm ứng dụng chạy trên hệ điều hành, firmware hoạt động ở cấp thấp, gần sát với phần cứng, và thường được lưu trữ trong bộ nhớ không thay đổi (ROM, EEPROM, Flash) của thiết bị.

Giả sử chúng ta sử dụng mạng có dây dùng công nghệ Ethernet. Card mạng Ethernet chứa chương trình firmware (tạm gọi là chương trình Ethernet) phù hợp để biến đổi dữ liệu thành tín hiệu điện (và ngược lại). Chương trình Ethernet sẽ tiếp nhận IP packet và chế tạo một chuỗi byte mới gọi là **frame** bằng cách thêm frame header (vào đầu) và frame trailer (vào sau IP packet).



Hình 3.7. Cấu trúc chuỗi byte header của frame

Frame là đơn vị dữ liệu giao thức cuối cùng được đóng gói ở dạng số. Card mạng sau đó sẽ chuyển frame thành tín hiệu và đưa lên đường truyền. Tuy nhiên, việc tạo frame Ethernet có một số vấn đề, phụ thuộc vào đích tới của dữ liệu. Vấn đề truyền frame trong LAN sẽ được xem xét chi tiết ở phần sau của giáo trình.

e) Khái niệm PDU, PCI, SDU

Ở trên chúng ta đã thấy, trong quá trình gọi dịch vụ của nhau, các chương trình trao đổi các gói tin tương tự như cách truyền tham số khi gọi hàm. Mỗi chương trình tạo ra gói tin riêng là một chuỗi byte có cấu trúc và được gọi chung là **PDU** (Protocol Data Unit - đơn vị dữ liệu của giao thức).

Mỗi PDU bao gồm:

- Đơn vị dữ liệu dịch vụ (Service Data Unit - **SDU**) là mảng byte dữ liệu chương trình gọi chuyển xuống.

- Thông tin điều khiển của giao thức (Protocol Control Information - **PCI**) là chuỗi byte do chương trình tạo ra chứa những thông tin cần thiết cho hoạt động của chương trình đó. Hai loại thông tin cần thiết nhất trong PCI là địa chỉ nguồn và địa chỉ đích. PCI cũng thường được gọi là **header**.

Như vậy, đơn vị dữ liệu giao thức là một chuỗi byte bao gồm chuỗi PCI gắn vào đầu chuỗi SDU, tức là PDU = PCI + SDU.

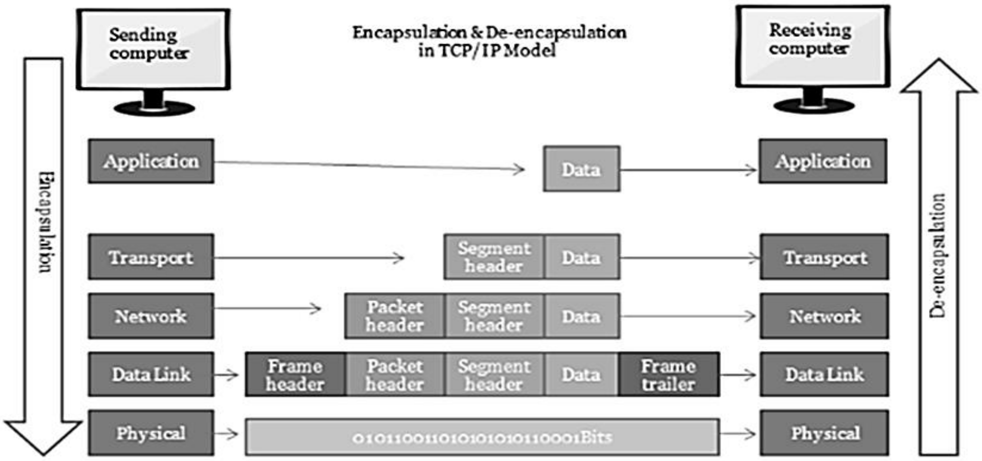
PDU của mỗi giao thức thường có tên gọi riêng:

- PDU của TCP được gọi là **segment**,
- PDU của UDP được gọi là **datagram**,
- PDU của IP được gọi là **packet**,
- PDU của Ethernet được gọi là **frame**.

f) Mở gói dữ liệu

Trên thiết bị đích, dữ liệu cũng trải qua một quá trình chạy ngược từ mạng tới ứng dụng, gọi là quá trình **mở gói dữ liệu**. Đối với thiết bị đầu cuối nhận dữ liệu, các bước diễn ra theo trình tự ngược lại so với ở thiết bị đầu cuối gửi dữ liệu. Cụ thể như sau:

1. Tín hiệu được card mạng chuyển thành frame.
2. Chương trình Ethernet cắt bỏ header và trailer của frame rồi chuyển phần SDU (chính là một IP packet) cho chương trình IP.
3. Chương trình IP cắt bỏ phần header để lấy riêng phần SDU trong packet (chính là TCP segment hoặc UDP dgram chứa trong IP packet) rồi chuyển cho TCP (hoặc UDP).
4. Chương trình gọi tới TCP/UDP để lấy chuỗi byte dữ liệu. Chương trình TCP/UDP tự động cắt bỏ phần header của mình để lấy riêng phần SDU chuyển cho chương trình.
5. Chương trình chuyển đổi chuỗi byte dữ liệu thành dữ liệu người dùng.


Sending computer
Encapsulation & De-encapsulation in TCP/IP Model
Receiving computer
Encapsulation
De-encapsulation
Application
Data
Application
Transport
Segment header
Data
Transport
Network
Packet header
Segment header
Data
Network
Data Link
Frame header
Packet header
Segment header
Data
Frame trailer
Data Link
Physical
010110011010101010110001Bits
Physical

Hình 3.8. Quy trình đóng / mở dữ liệu với TCP

3.3. Thực hành - bắt và xem cấu trúc dữ liệu với Wireshark

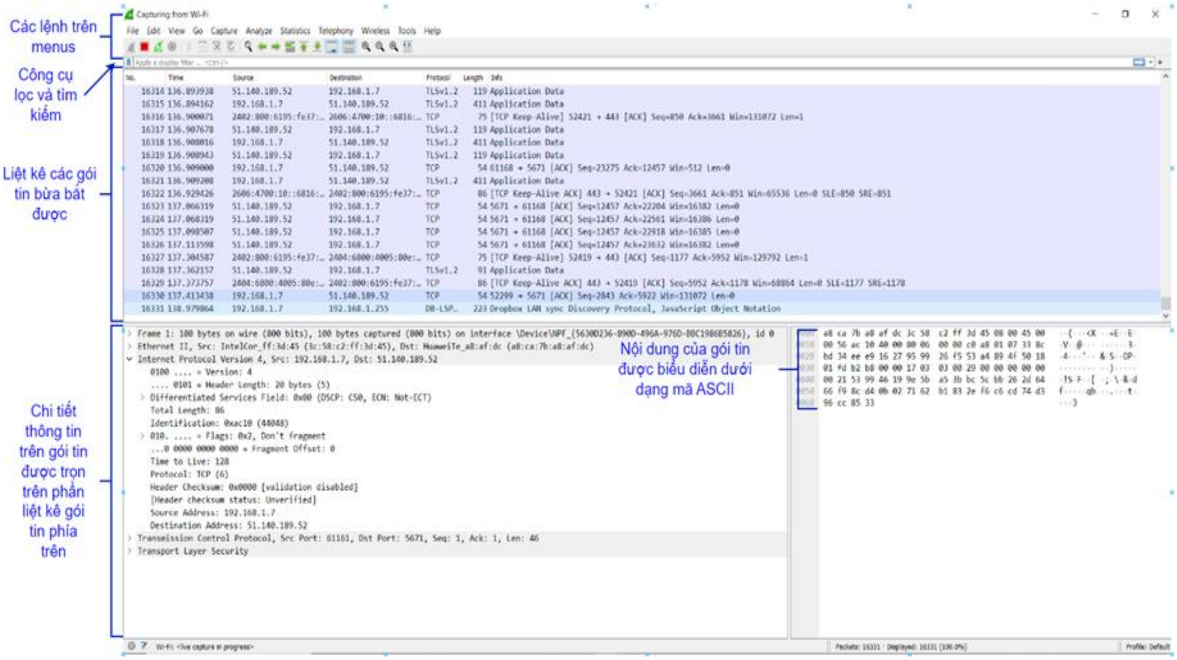
3.3.1. Giới thiệu Wireshark

Wireshark là một công cụ phân tích và giám sát mạng mạnh mẽ trong môi trường máy tính. Nó cung cấp khả năng theo dõi, ghi lại và phân tích giao thức mạng, cho phép người dùng hiểu và giải quyết các vấn đề liên quan đến mạng.

Ứng dụng của Wireshark trong mạng máy tính bao gồm:

- Phân tích và sửa lỗi mạng: Wireshark cung cấp khả năng theo dõi và ghi lại các gói tin trên mạng. Điều này cho phép người dùng kiểm tra lỗi mạng, tìm ra nguyên nhân gây ra vấn đề và sửa chữa nhanh chóng. Wireshark cung cấp khả năng phân tích các giao thức mạng như TCP, UDP, ICMP, DNS, DHCP và nhiều giao thức khác, giúp người dùng hiểu rõ hơn về hoạt động của mạng và xác định các vấn đề liên quan.
- Kiểm tra hiệu suất mạng: Wireshark cho phép người dùng theo dõi và đo lường hiệu suất mạng. Bằng cách phân tích các thời gian trễ, số lượng gói tin gửi và nhận, người dùng có thể đánh giá và tối ưu hóa hiệu suất mạng. Wireshark cung cấp thông tin chi tiết về tốc độ truyền dẫn, lưu lượng mạng và tình trạng kết nối, giúp người dùng xác định các vấn đề liên quan đến hiệu suất và tăng cường khả năng chịu tải của mạng.
- Giám sát bảo mật mạng: Wireshark có thể được sử dụng để giám sát và phân tích lưu lượng mạng để phát hiện các hoạt động không đáng tin cậy hoặc các mối đe dọa bảo mật. Bằng cách theo dõi các giao thức mạng và xem xét các gói tin, người dùng có thể xác định các tấn công mạng như tin tặc, xâm nhập, spoofing địa chỉ IP và các hoạt động không được ủy quyền khác. Điều này giúp cải thiện bảo mật mạng và thực hiện các biện pháp phòng ngừa.
- Phân tích giao thức mạng: Wireshark cung cấp khả năng phân tích các giao thức mạng chi tiết. Người dùng có thể xem và theo dõi các trường dữ liệu trong các gói tin, xác định các lỗi hoặc không tuân thủ trong giao thức, và phân tích quy trình truyền tin nhắn. Điều này giúp hiểu rõ hơn về hoạt động của các giao thức mạng và giải quyết các vấn đề liên quan.
- Nghiên cứu về mạng: Wireshark cho phép người dùng đi sâu vào các vấn đề mạng và tìm hiểu rõ hơn về hoạt động của mạng. Người dùng có thể xem xét các thông tin như các trường dữ liệu, kích thước gói tin, giao thức kết nối và các đặc điểm khác để xác định nguyên nhân gốc rễ của các vấn đề mạng phức tạp.

Như vậy có thể thấy, phần mềm Wireshark là một công cụ quan trọng và mạnh mẽ trong mạng máy tính. Nó cung cấp khả năng phân tích, giám sát và ghi lại lưu lượng mạng, giúp người dùng tìm ra nguyên nhân và giải quyết các vấn đề mạng, cải thiện hiệu suất và bảo mật của mạng.


Các lệnh trên menus
Công cụ lọc và tìm kiếm
Liệt kê các gói tin bừa bắt được
Chi tiết thông tin trên gói tin được trọn trên phần liệt kê gói tin phía trên
Nội dung của gói tin được biểu diễn dưới dạng mã ASCII
Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Wi-Fi: <live capture in progress>
Packets: 16331 · Displayed: 16331 (100.0%)
Profile: Default

```
No.   Time        Source              Destination         Protocol Length Info
16314 136.893938  51.140.189.52       192.168.1.7         TLSv1.2  119 Application Data
16315 136.894162  192.168.1.7         51.140.189.52       TLSv1.2  411 Application Data
16316 136.900071  2402:800:6195:fe37:… 2606:4700:10::6816:… TCP    75 [TCP Keep-Alive] 52421 → 443 [ACK] Seq=850 Ack=3661 Win=131072 Len=1
16317 136.907678  51.140.189.52       192.168.1.7         TLSv1.2  119 Application Data
16318 136.908016  192.168.1.7         51.140.189.52       TLSv1.2  411 Application Data
16319 136.908943  51.140.189.52       192.168.1.7         TLSv1.2  119 Application Data
16320 136.909000  192.168.1.7         51.140.189.52       TCP       54 61168 → 5671 [ACK] Seq=23275 Ack=12457 Win=512 Len=0
16321 136.909208  192.168.1.7         51.140.189.52       TLSv1.2  411 Application Data
16322 136.929426  2606:4700:10::6816:… 2402:800:6195:fe37:… TCP    86 [TCP Keep-Alive ACK] 443 → 52421 [ACK] Seq=3661 Ack=851 Win=65536 Len=0 SLE=850 SRE=851
16323 137.066319  51.140.189.52       192.168.1.7         TCP       54 5671 → 61168 [ACK] Seq=12457 Ack=22204 Win=16382 Len=0
16324 137.068319  51.140.189.52       192.168.1.7         TCP       54 5671 → 61168 [ACK] Seq=12457 Ack=22561 Win=16386 Len=0
16325 137.098507  51.140.189.52       192.168.1.7         TCP       54 5671 → 61168 [ACK] Seq=12457 Ack=22918 Win=16385 Len=0
16326 137.113598  51.140.189.52       192.168.1.7         TCP       54 5671 → 61168 [ACK] Seq=12457 Ack=23632 Win=16382 Len=0
16327 137.304587  2402:800:6195:fe37:… 2404:6800:4005:80e:… TCP    75 [TCP Keep-Alive] 52419 → 443 [ACK] Seq=1177 Ack=5952 Win=129792 Len=1
16328 137.362157  51.140.189.52       192.168.1.7         TLSv1.2   91 Application Data
16329 137.373757  2404:6800:4005:80e:… 2402:800:6195:fe37:… TCP    86 [TCP Keep-Alive ACK] 443 → 52419 [ACK] Seq=5952 Ack=1178 Win=68864 Len=0 SLE=1177 SRE=1178
16330 137.413438  192.168.1.7         51.140.189.52       TCP       54 52299 → 5671 [ACK] Seq=2843 Ack=5922 Win=131072 Len=0
16331 138.979864  192.168.1.7         192.168.1.255       DB-LSP…  223 Dropbox LAN sync Discovery Protocol, JavaScript Object Notation
```

```
> Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface \Device\NPF_{5630D236-890D-496A-976D-80C198685826}, id 0
> Ethernet II, Src: IntelCor_ff:3d:45 (3c:58:c2:ff:3d:45), Dst: HuaweiTe_a8:af:dc (a8:ca:7b:a8:af:dc)
v Internet Protocol Version 4, Src: 192.168.1.7, Dst: 51.140.189.52
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 86
    Identification: 0xac10 (44048)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.7
    Destination Address: 51.140.189.52
> Transmission Control Protocol, Src Port: 61161, Dst Port: 5671, Seq: 1, Ack: 1, Len: 46
> Transport Layer Security
```

```
0000  a8 ca 7b a8 af dc 3c 58  c2 ff 3d 45 08 00 45 00   ··{···<X ··=E··E·
0010  00 56 ac 10 40 00 80 06  00 00 c0 a8 01 07 33 8c   ·V··@··· ······3·
0020  bd 34 ee e9 16 27 95 99  26 f5 53 a4 89 4f 50 18   ·4···'·· &·S··OP·
0030  01 fd b2 b8 00 00 17 03  03 00 29 00 00 00 00 00   ········ ··)·····
0040  00 21 53 99 46 19 9e 5b  a5 3b bc 5c bb 26 2d 64   ·!S·F··[ ·;·\·&-d
0050  66 f9 8c d4 0b 02 71 62  b1 83 2e f6 c6 cd 74 d3   f·····qb ··.···t·
0060  96 cc 85 33                                        ···3
```

Hình 3.9. Ý nghĩa của các phần thông tin trên cửa sổ bắt gói tin

3.3.2. Sử dụng Wireshark để bắt và xem gói tin

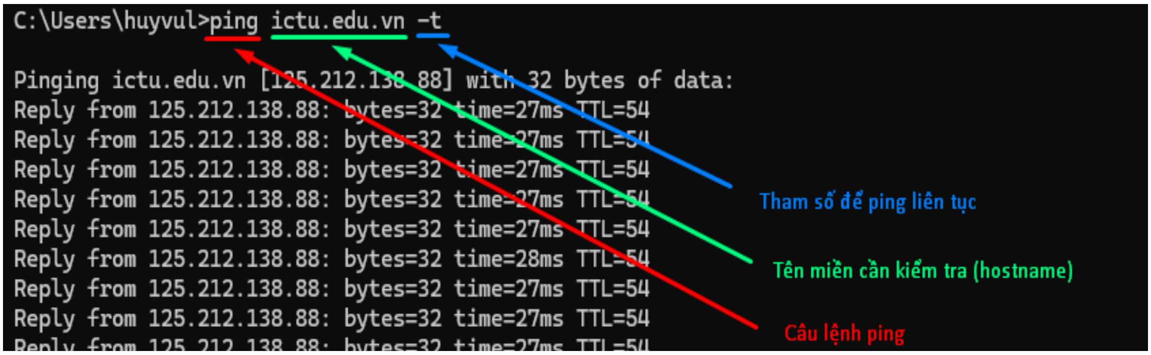
a) Bắt và xem gói tin ping từ máy tính

Chương trình Ping là một công cụ đơn giản cho phép mỗi người có thể kiểm tra xem host của mình còn hoạt động hay không. Chương trình Ping trong máy nguồn gửi một gói tin đến địa chỉ IP đích. Nếu đích đó vẫn hoạt động, chương trình Ping trong máy đích phản hồi lại bằng việc gửi một gói tin về cho máy nguồn.

(1) Mở ứng dụng Command Prompt

(2) Khởi động Wireshark và bắt đầu bắt gói tin

(3) Lệnh ping ở trong Command Prompt

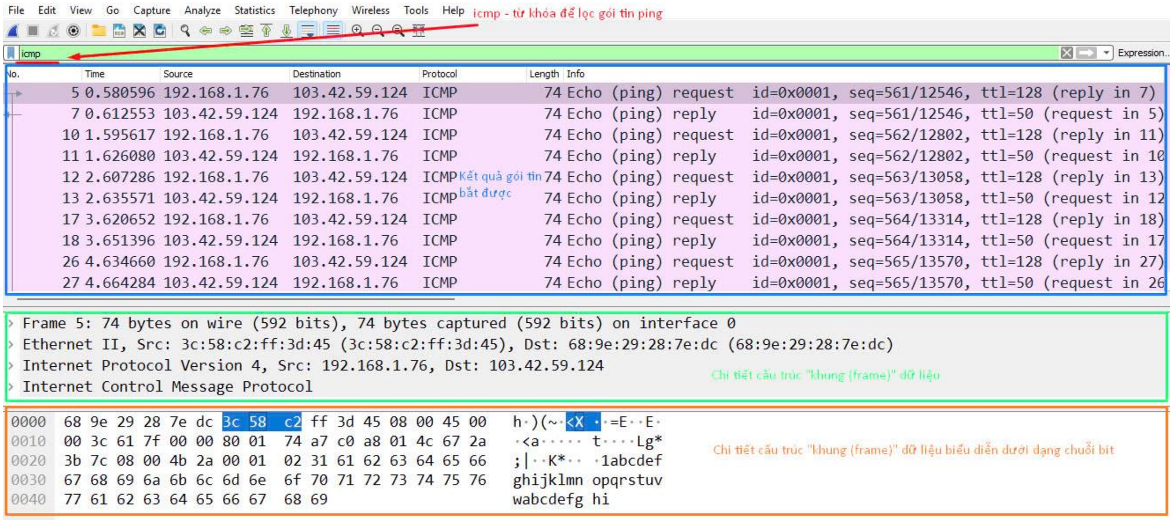


```
C:\Users\huyvul>ping ictu.edu.vn -t

Pinging ictu.edu.vn [125.212.138.88] with 32 bytes of data:
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=28ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
Reply from 125.212.138.88: bytes=32 time=27ms TTL=54
```

Tham số để ping liên tục
Tên miền cần kiểm tra (hostname)
Câu lệnh ping

Kết quả chạy lệnh ping ictu.edu.vn -t

(4) Xem kết quả


File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
icmp - từ khóa để lọc gói tin ping
Expression...
Kết quả gói tin bắt được
Chi tiết cấu trúc "khung (frame)" dữ liệu
Chi tiết cấu trúc "khung (frame)" dữ liệu biểu diễn dưới dạng chuỗi bit

```
icmp
```

```
No.     Time     Source         Destination    Protocol  Length Info
      5 0.580596 192.168.1.76   103.42.59.124  ICMP      74 Echo (ping) request  id=0x0001, seq=561/12546, ttl=128 (reply in 7)
      7 0.612553 103.42.59.124  192.168.1.76   ICMP      74 Echo (ping) reply    id=0x0001, seq=561/12546, ttl=50 (request in 5)
     10 1.595617 192.168.1.76   103.42.59.124  ICMP      74 Echo (ping) request  id=0x0001, seq=562/12802, ttl=128 (reply in 11)
     11 1.626080 103.42.59.124  192.168.1.76   ICMP      74 Echo (ping) reply    id=0x0001, seq=562/12802, ttl=50 (request in 10
     12 2.607286 192.168.1.76   103.42.59.124  ICMP      74 Echo (ping) request  id=0x0001, seq=563/13058, ttl=128 (reply in 13)
     13 2.635571 103.42.59.124  192.168.1.76   ICMP      74 Echo (ping) reply    id=0x0001, seq=563/13058, ttl=50 (request in 12
     17 3.620652 192.168.1.76   103.42.59.124  ICMP      74 Echo (ping) request  id=0x0001, seq=564/13314, ttl=128 (reply in 18)
     18 3.651396 103.42.59.124  192.168.1.76   ICMP      74 Echo (ping) reply    id=0x0001, seq=564/13314, ttl=50 (request in 17
     26 4.634660 192.168.1.76   103.42.59.124  ICMP      74 Echo (ping) request  id=0x0001, seq=565/13570, ttl=128 (reply in 27)
     27 4.664284 103.42.59.124  192.168.1.76   ICMP      74 Echo (ping) reply    id=0x0001, seq=565/13570, ttl=50 (request in 26
```

```
> Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: 3c:58:c2:ff:3d:45 (3c:58:c2:ff:3d:45), Dst: 68:9e:29:28:7e:dc (68:9e:29:28:7e:dc)
> Internet Protocol Version 4, Src: 192.168.1.76, Dst: 103.42.59.124
> Internet Control Message Protocol
```

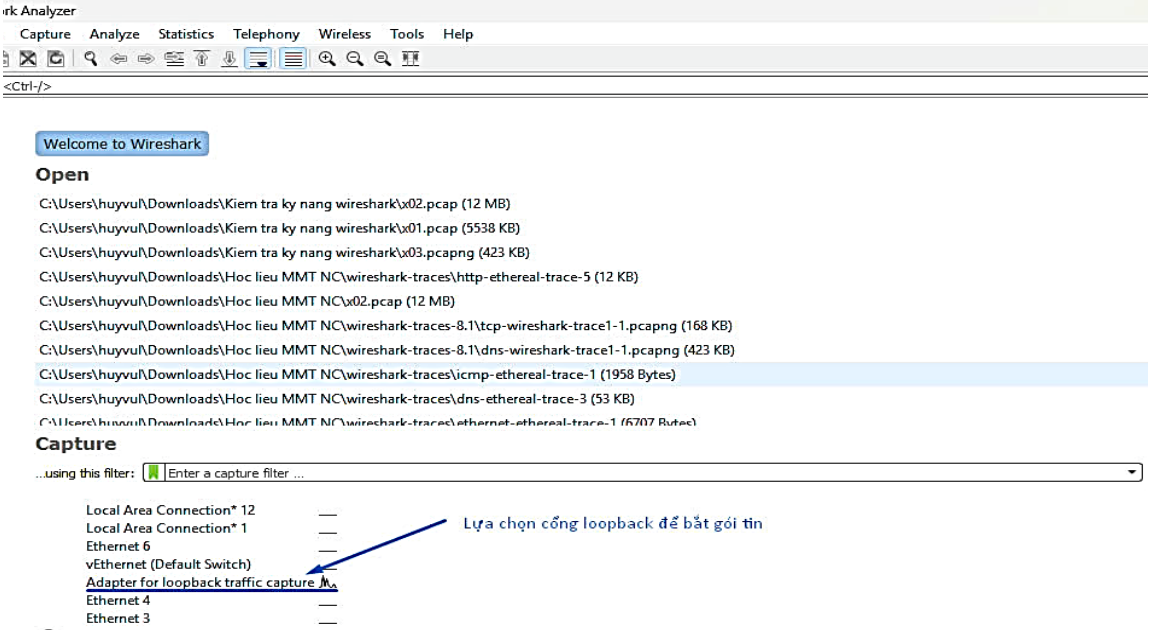
```
0000  68 9e 29 28 7e dc 3c 58  c2 ff 3d 45 08 00 45 00   h·)(~·<X ··=E··E·
0010  00 3c 61 7f 00 00 80 01  74 a7 c0 a8 01 4c 67 2a   ·<a····· t····Lg*
0020  3b 7c 08 00 4b 2a 00 01  02 31 61 62 63 64 65 66   ;|··K*·· ·1abcdef
0030  67 68 69 6a 6b 6c 6d 6e  6f 70 71 72 73 74 75 76   ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67  68 69                     wabcdefg hi
```

Kết quả lọc các gói tin ICMP

b) Bắt và xem gói tin UDP

Sử dụng chương trình gửi/nhận gói tin (client/server) trên nên tảng giao thức UDP đã được giới thiệu ở mục 3 để gửi và nhận một chuỗi ký tự ngẫu nhiên sau đó dùng wireshark để bắt gói tin vừa được gửi từ client tới server. Mục tiêu của bài thực hành là quan sát dữ liệu được biến đổi dưới dạng bits/bytes và kết quả của quá trình đóng gói dữ liệu để gói có thể gửi đi từ máy client đến máy server.

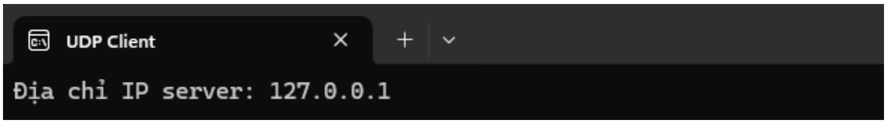
(1) Mở phần mềm Wireshark, chọn loopback như là interface cần thu thập gói tin


rk Analyzer
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
<Ctrl-/>
Welcome to Wireshark
Open
Capture
...using this filter:
Enter a capture filter ...
Local Area Connection* 12
Local Area Connection* 1
Ethernet 6
vEthernet (Default Switch)
Adapter for loopback traffic capture
Ethernet 4
Ethernet 3
Lựa chọn cổng loopback để bắt gói tin

C:\Users\huyvul\Downloads\Kiem tra ky nang wireshark\x02.pcap (12 MB)
C:\Users\huyvul\Downloads\Kiem tra ky nang wireshark\x01.pcap (5538 KB)
C:\Users\huyvul\Downloads\Kiem tra ky nang wireshark\x03.pcapng (423 KB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces\http-ethereal-trace-5 (12 KB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\x02.pcap (12 MB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces-8.1\tcp-wireshark-trace1-1.pcapng (168 KB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces-8.1\dns-wireshark-trace1-1.pcapng (423 KB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces\icmp-ethereal-trace-1 (1958 Bytes)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces\dns-ethereal-trace-3 (53 KB)
C:\Users\huyvul\Downloads\Hoc lieu MMT NC\wireshark-traces\ethernet-ethereal-trace-1 (6707 Bytes)

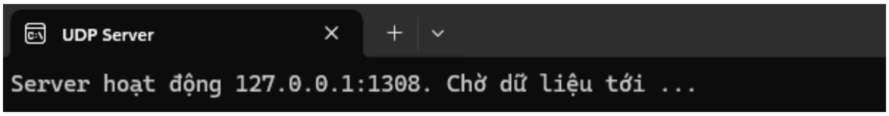
Lựa chọn giao diện loopback để bắt gói tin

(2) Mở chương trình client, chọn địa chỉ ip dùng để truyền nhận tin là 127.0.0.1



```
Địa chỉ IP server: 127.0.0.1
```

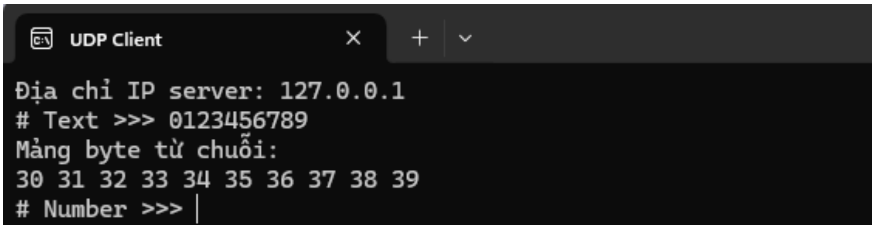
(3) Biên dịch và chạy chương trình server, chương trình sẽ đợi nhận tin từ địa chỉ Ip 127.0.0.1:



```
Server hoạt động 127.0.0.1:1308. Chờ dữ liệu tới ...
```

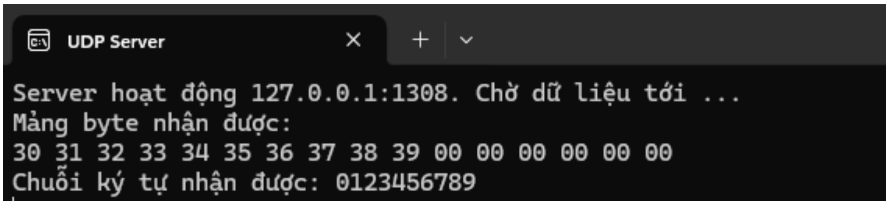
(4) Gửi chuỗi ký tự 0123456789 từ client tới server

Client gửi:



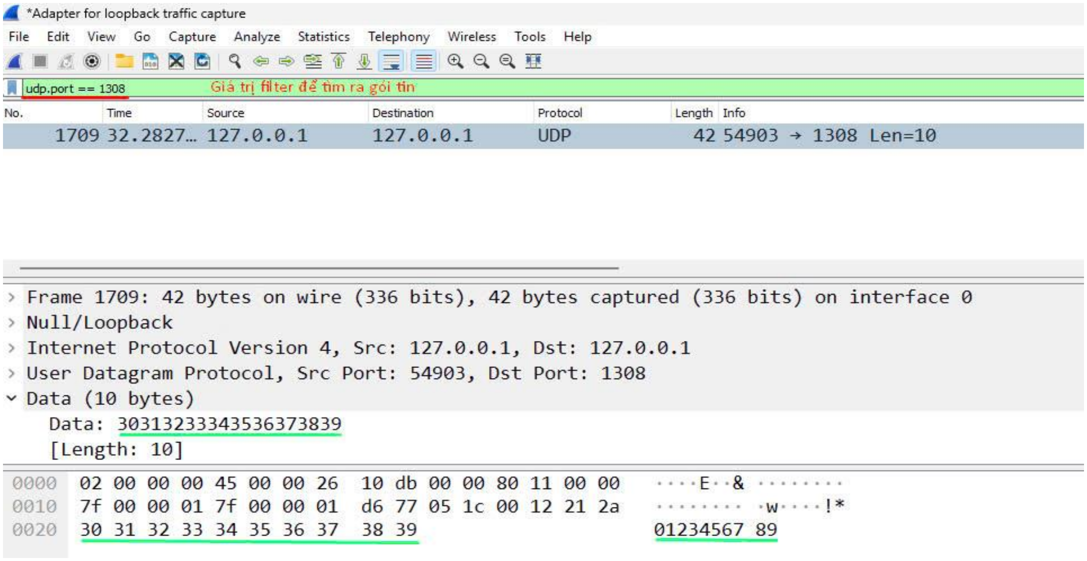
```
Địa chỉ IP server: 127.0.0.1
# Text >>> 0123456789
Mảng byte từ chuỗi:
30 31 32 33 34 35 36 37 38 39
# Number >>> 
```

Server nhận:



```
Server hoạt động 127.0.0.1:1308. Chờ dữ liệu tới ...
Mảng byte nhận được:
30 31 32 33 34 35 36 37 38 39 00 00 00 00 00 00
Chuỗi ký tự nhận được: 0123456789
```

Wireshark bắt được:


*Adapter for loopback traffic capture
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help

udp.port == 1308

Giá trị filter để tìm ra gói tin

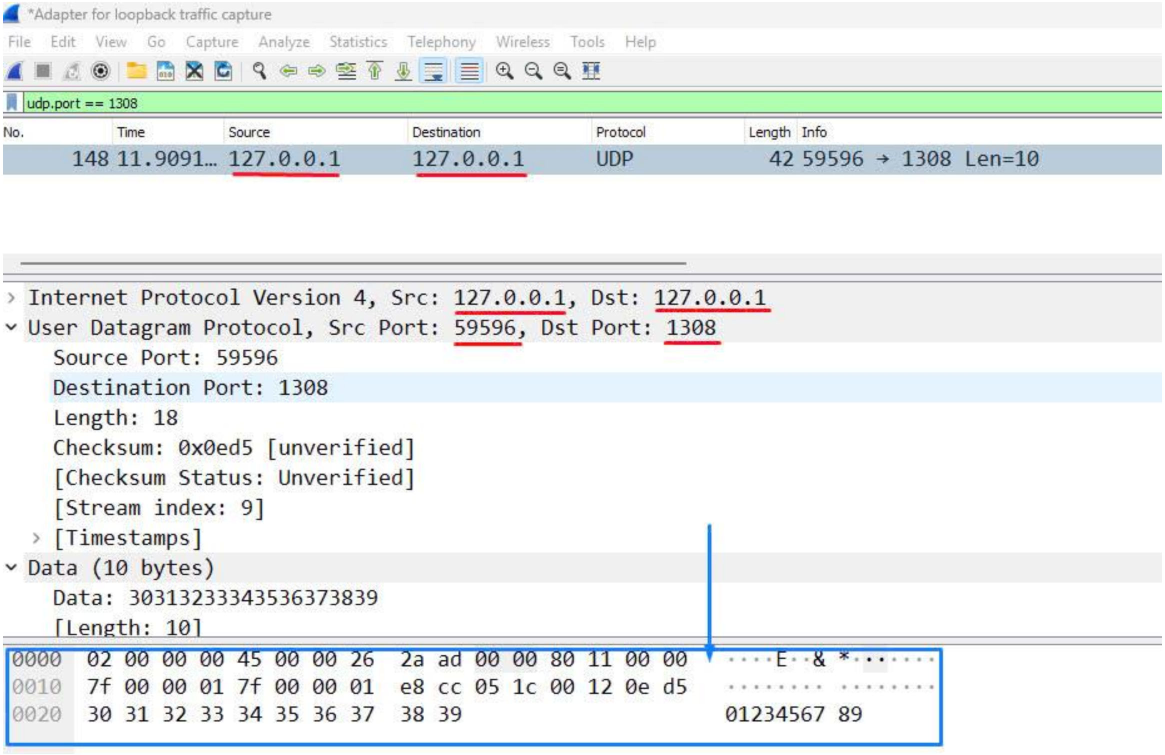
```
No.        Time        Source       Destination   Protocol   Length Info
    1709 32.2827… 127.0.0.1    127.0.0.1     UDP            42 54903 → 1308 Len=10

> Frame 1709: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 54903, Dst Port: 1308
˅ Data (10 bytes)
    Data: 30313233343536373839
    [Length: 10]

0000  02 00 00 00 45 00 00 26  10 db 00 00 80 11 00 00   ····E··& ········
0010  7f 00 00 01 7f 00 00 01  d6 77 05 1c 00 12 21 2a   ········ ·w····!*
0020  30 31 32 33 34 35 36 37  38 39                     01234567 89
```

Kết quả bắt gói tin UDP do chương trình phát ra

Để lọc đúng gói tin cần xem, trong phần filter điền giá trị: udp.port==1308

Chuỗi 0123456789 được biến đổi thành 0x30313233343536373839 (biểu diễn hexa)

Có thể thấy để gửi được chuỗi 0123456789 qua mạng, diễn ra quá trình đóng gói dữ liệu tại máy gửi (trong trường hợp này máy client thêm địa chỉ IP, thêm số hiệu cổng....) sau đó dữ liệu được chuyển qua dạng chuỗi các bít để chuyển đến máy nhận (server trong trường hợp này). Chuỗi bít sẽ được xử lý loại bỏ các phần dữ liệu header, giữ lại phần bít thông tin dữ liệu (quá trình mở gói dữ liệu) sau đó hiển thị chuỗi bít thông tin dữ liệu thành chuỗi ban đầu (0123456789) trên cửa sổ hiển thị của chương trình server.


*Adapter for loopback traffic capture
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 1308

```
No.   Time          Source      Destination   Protocol   Length  Info
  148 11.9091…      127.0.0.1   127.0.0.1     UDP            42  59596 → 1308 Len=10
```

```
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
˅ User Datagram Protocol, Src Port: 59596, Dst Port: 1308
    Source Port: 59596
    Destination Port: 1308
    Length: 18
    Checksum: 0x0ed5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 9]
  > [Timestamps]
˅ Data (10 bytes)
    Data: 30313233343536373839
    [Length: 10]
```

```
0000  02 00 00 00 45 00 00 26  2a ad 00 00 80 11 00 00   ····E··& *·······
0010  7f 00 00 01 7f 00 00 01  e8 cc 05 1c 00 12 0e d5   ········ ········
0020  30 31 32 33 34 35 36 37  38 39                     01234567 89
```

Hình 3.10. Kết quả bắt gói tin UDP trong Wireshark

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) Hãy sử dụng mã nguồn đã hoàn thành ở mục 3.1.1. Chạy chương trình client và server, thực hiện truyền một số từ client đến server sau đó bắt và quan sát gói tin vừa bắt được? Dữ liệu thu được khi bắt gói tin truyền một số có gì khác so với dữ liệu thu được khi truyền một chuỗi ký tự? Giải thích tại sao?

2) Hãy sử dụng mã nguồn đã hoàn thành ở mục 3.1.1. Chạy chương trình client và server, thực hiện truyền một chuỗi 200 ký tự từ client đến server sau đó bắt và quan sát gói tin vừa bắt được? Có gì khác biệt so với việc bắt chuỗi ký tự ngắn 0123456789 bài thực hành trước?

Câu hỏi lý thuyết

1) Khi một ứng dụng muốn gửi một chuỗi văn bản qua mạng, vì sao chuỗi này cần được biến đổi thành mảng byte? Nếu không thực hiện bước này, điều gì có thể xảy ra?
2) Giả sử bạn cần gửi số thực kiểu float qua mạng, tại sao không thể gửi trực tiếp như cách hiển thị trên màn hình? Việc biến đổi kiểu dữ liệu này cần chú ý điều gì?
3) Trong quá trình truyền dữ liệu, vì sao cần đóng gói chuỗi byte thành dạng có cấu trúc (ví dụ: thêm độ dài, nhãn phân loại)? Điều này giúp ích gì cho thiết bị nhận?
4) Hãy suy luận: nếu dữ liệu được gửi đi mà không có thông tin báo trước về kích thước, thiết bị nhận sẽ gặp khó khăn gì khi đọc dữ liệu?
5) Nếu bạn muốn gửi nhiều loại dữ liệu (ví dụ: chuỗi ký tự, số nguyên, số thực) trong cùng một lần truyền, bạn cần tổ chức chuỗi byte theo nguyên tắc nào để thiết bị nhận hiểu đúng từng phần?
6) Trong thực tế, thứ tự byte khi truyền dữ liệu giữa các máy tính có thể khác nhau. Điều gì sẽ xảy ra nếu không chuẩn hóa thứ tự này trước khi gửi?
7) Tại sao cần có cơ chế chuyển đổi giữa kiểu dữ liệu cấp cao (như chuỗi, số) và mảng byte khi truyền dữ liệu giữa hai máy? Việc này có cần thiết ở cả hai đầu gửi và nhận không? Giải thích.
8) Hãy tưởng tượng bạn đang thiết kế phần mềm gửi dữ liệu dạng văn bản từ máy tính A sang máy tính B. Bạn cần thực hiện những bước xử lý dữ liệu nào trước khi truyền?
9) Giả sử bạn viết chương trình để gửi tệp ảnh qua mạng. Ảnh là dữ liệu nhị phân, vậy cách đóng gói và giải mã dữ liệu ảnh sẽ khác gì so với văn bản?

BÀI 4. GIAO THỨC VÀ BỘ GIAO THỨC

Mục tiêu của bài học

Bài học giúp sinh viên hiểu rõ khái niệm giao thức truyền thông mạng, vai trò và nguyên lý hoạt động của các loại giao thức phổ biến trong mạng máy tính, đồng thời rèn luyện kỹ năng lập trình truyền dữ liệu cơ bản qua TCP. Cụ thể, sau khi hoàn thành bài học, sinh viên có thể:

Về kiến thức:

- Mô tả được khái niệm giao thức mạng và các thành phần chính của một giao thức.
- Phân biệt giữa đặc tả và thực thi giao thức, giữa giao thức mở và giao thức độc quyền.
- Giải thích được các loại dịch vụ truyền thông như end-to-end, host-to-host, tin cậy/không tin cậy, hướng liên kết/phi liên kết.
- Trình bày được cấu trúc tổng quan của bộ giao thức TCP/IP và chức năng của các giao thức cơ bản như TCP, UDP, IP.

Về kỹ năng:

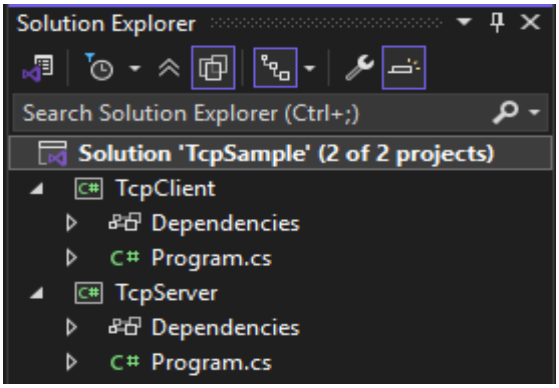
- Cài đặt thành công một chương trình client-server đơn giản sử dụng giao thức TCP.
- Phân tích được dữ liệu truyền qua mạng bằng công cụ Wireshark, bao gồm xác định cổng nguồn/đích và nội dung payload.
- Sử dụng kết hợp kiến thức về số cổng và địa chỉ IP để mô hình hóa địa chỉ điểm cuối trong truyền thông mạng.

4.1. Thực hành - Viết chương trình đơn giản cho phép truyền dữ liệu với TCP

Trong bài thực hành này chúng ta sẽ cùng xây dựng một phần mềm đơn giản để minh họa cho việc thiết kế và thực thi một giao thức.

Bước 1: Tạo một Solution trống đặt tên là TcpSample

Bước 2: Xây dựng lần lượt hai dự án thuộc kiểu Console App trong solution trên với tên gọi lần lượt là Server và Client.


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'TcpSample' (2 of 2 projects)
TcpClient
Dependencies
Program.cs
TcpServer
Dependencies
Program.cs

Hình 4.1. Cấu trúc solution với hai dự án, TcpClient và TcpServer

Bước 3: Thiết lập để có thể debug đồng thời cả hai chương trình, trong đó Server sẽ khởi động trước.

Bước 4: Mở file Program.cs của Client và viết code như sau:

Client / Program.cs

```
using System.Net.Sockets;
using System.Net;
using System.Text;
using static System.Console;

Title = "TCP Client";
// yêu cầu người dùng nhập ip của server
Write("Server IP address: ");
var serverIpStr = ReadLine();
// chuyển đổi chuỗi ký tự thành object thuộc kiểu IPAddress
var serverIp = IPAddress.Parse(serverIpStr);
var serverPort = 1308;
// đây là "địa chỉ" của tiến trình server trên mạng
// mỗi endpoint chứa ip của host và port của tiến trình
var serverEndpoint = new IPEndPoint(serverIp, serverPort);
var size = 1024; // kích thước của bộ đệm
var receiveBuffer = new byte[size]; // mảng byte làm bộ đệm
while (true) {
    // yêu cầu người dùng nhập một chuỗi
    Write("# Text >>> ");
    ResetColor();
    var text = ReadLine();
    // khởi tạo object của lớp socket để sử dụng dịch vụ TCP
    var socket = new Socket(SocketType.Stream, ProtocolType.Tcp);
    socket.Connect(serverEndpoint);
    // biến đổi chuỗi thành mảng byte
    var sendBuffer = Encoding.ASCII.GetBytes(text);
    // gửi mảng byte trên đến tiến trình server
    socket.Send(sendBuffer);
    // endpoint này chỉ dùng khi nhận dữ liệu
    socket.Close();
}
```

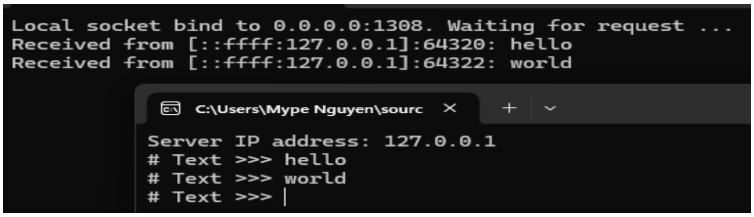
Bước 5: Mở file Program.cs của Server và viết code như sau:

Server / Program.cs

```
using System.Net.Sockets;
using System.Net;
using System.Text;
using static System.Console;

Title = "TCP Server";
// giá trị Any của IPAddress tương ứng với IP của tất cả các giao diện mạng trên máy
var localIp = IPAddress.Any;
// tiến trình server sẽ sử dụng cổng 1308
var localPort = 1308;
// biến này sẽ chứa "địa chỉ" của tiến trình server trên mạng
var localEndPoint = new IPEndPoint(localIp, localPort);
// yêu cầu hệ điều hành cho phép chiếm dụng cổng 1308
// server sẽ nghe trên tất cả các mạng mà máy tính này kết nối tới
var listener = new Socket(SocketType.Stream, ProtocolType.Tcp);
listener.Bind(localEndPoint);
// bắt đầu lắng nghe các yêu cầu từ client
listener.Listen(10);
WriteLine($"Local socket bind to {localEndPoint}. Waiting for request ...");
var size = 1024;
var receiveBuffer = new byte[size];
while (true) {
   var socket = listener.Accept();
   var length = socket.Receive(receiveBuffer);
   var text = Encoding.ASCII.GetString(receiveBuffer, 0, length);
   WriteLine($"Received from {socket.RemoteEndPoint}: {text}");
   socket.Close();
}
```

Bước 6: Chạy thử ứng dụng bằng phím F5 hoặc tổ hợp Ctrl+F5. Khi này, chương trình server sẽ chạy trước, chương trình client sẽ chạy ngay sau server.

Nhập bất kỳ chuỗi ký tự nào vào dấu nhắc lệnh của client và ấn Enter, chuỗi ký tự này sẽ được gửi tới chương trình Server.



```
Local socket bind to 0.0.0.0:1308. Waiting for request ...
Received from [::ffff:127.0.0.1]:64320: hello
Received from [::ffff:127.0.0.1]:64322: world
```

```
Server IP address: 127.0.0.1
# Text >>> hello
# Text >>> world
# Text >>> |
```

Hình 4.2. Kết quả hoạt động của chương trình truyền thông tin qua TCP

4.2. Giao thức và dịch vụ truyền thông mạng

4.2.1. Khái niệm

a) Giao thức mạng

Giao thức mạng là tập hợp các quy tắc điều khiển quá trình truyền thông mà các thành phần của hệ thống mạng phải tuân theo khi tham gia vào quá trình truyền thông. Mỗi giao thức, nhìn giản lược, là một bộ quy tắc quy định 3 yếu tố chính: cấu trúc của các thông tin trao đổi, trình tự trao đổi thông tin, các hành động cần thực hiện khi gửi/nhận thông tin. Trong một hệ thống mạng đồng thời có rất nhiều giao thức hoạt động để đảm bảo cho việc truyền thông tin giữa các thành phần.

Cụ thể hơn, một giao thức thường xác định:

- Các loại thông điệp được trao đổi: ví dụ, thông điệp truy vấn (request) và thông điệp phản hồi (response).
- Cấu trúc và cú pháp của các thông điệp: ví dụ, danh sách các trường trong từng loại thông điệp và cách thức tạo ra các trường đó.
- Ý nghĩa và cách diễn giải của các trường dữ liệu: giá trị và vai trò của thông tin trong các trường.
- Thứ tự và điều kiện gửi / nhận thông điệp.

Giao thức truyền thông đóng vai trò một ngôn ngữ chung gắn kết và phối hợp hoạt động của các thiết bị về mặt logic (để phân biệt với môi trường truyền gắn kết thiết bị về mặt vật lý). Do có nhiều thành phần khác nhau trong mạng, mỗi thành phần có vai trò riêng, giao thức là bắt buộc để các thành phần có được "tiếng nói chung" nhằm cùng thực hiện nhiệm vụ. Như vậy, giao thức truyền thông đóng vai trò là một ngôn ngữ chung để các thiết bị có thể "nói chuyện" và hợp tác với nhau trong một hệ thống mạng.

Ứng với mỗi loại nhiệm vụ truyền thông sẽ phải có một/một số giao thức truyền thông tương ứng. Ví dụ, giao thức dùng để truyền thông tin trực tiếp giữa các ứng dụng, giao thức truyền thông tin giữa các thiết bị đầu cuối (và có thể được sử dụng bởi phần mềm ứng dụng), giao thức dùng để trao đổi thông tin giữa các thiết bị mạng, v.v..

Việc triển khai và thực thi giao thức là cần thiết trong cả phần cứng và phần mềm mạng.

Đối với phần cứng, ví dụ, các router sử dụng các giao thức định tuyến để tìm ra đường đi phù hợp cho các gói tin xuyên qua các mạng khác nhau. Các switch cũng phải sử dụng giao thức riêng để học địa MAC của các thiết bị kết nối vào các cổng của nó. Một thiết bị thu phát wifi (Access Point) sử dụng giao thức riêng để truyền tín hiệu với máy tính xách tay sử dụng mạng không dây.

Đối với phần mềm, các giao thức được sử dụng cũng rất đa dạng. Mỗi phần mềm đều phải sử dụng một hoặc một số giao thức riêng. Ví dụ, trong hệ thống phần mềm web, trình duyệt sử dụng giao thức HTTP để nói chuyện với chương trình máy chủ. Chương trình email client sử dụng các giao thức SMTP, POP3, IMAP để nói chuyện với chương trình máy chủ email.

b) Đặc tả và thực thi giao thức

Theo định nghĩa ở trên, giao thức có thể hiểu là một bản mô tả các quy tắc chi phối quá trình truyền thông tin trong mạng, bao gồm: định dạng của các thông tin được trao đổi giữa các thành phần trong mạng, các hoạt động phải thực hiện khi gửi/nhận thông tin, thứ tự thực hiện gửi / nhận thông tin. Giao thức, theo ý nghĩa này, chỉ, đơn thuần là một bản mô tả các quy tắc và được viết thành văn bản, gọi là đặc tả của giao thức.

Các nhà phát triển thiết bị, hệ thống và ứng dụng sử dụng bản đặc tả này cùng với công cụ lập trình để chuyển đặc tả giao thức thành chương trình và cài đặt lên các thiết bị. Loại chương trình như vậy được gọi là thực thi của giao thức. Chỉ khi được thực thi và cài đặt, giao thức mới có thể thực sự hoạt động trong hệ thống mạng.

Như vậy, đặc tả của giao thức có thể xem như một dạng bản thiết kế, còn thực thi của giao thức chính là một chương trình được làm ra từ bản thiết kế.

Bình thường khi nói về mạng chúng ta không cần phân biệt rạch ròi giữa đặc tả và thực thi của giao thức. Tuy nhiên, tùy tình huống cụ thể chúng ta sẽ phải hiểu và phân biệt hai khái niệm trên.

Với sự phân biệt như vậy chúng ta có thể hình dung, thực thi của một giao thức thực chất là một chương trình. Trên thiết bị đầu cuối, chương trình thực thi của giao thức có thể được tích hợp trong phần mềm ứng dụng, cài đặt trên hệ điều hành, hoặc cài đặt trực tiếp trên phần cứng (như card mạng). Trên thiết bị mạng, chương trình thực thi của giao thức được cài đặt và vận hành trực tiếp trên phần cứng hoặc hệ điều hành riêng của thiết bị.

c) Giao thức mở và giao thức độc quyền

Khi phát triển một ứng dụng hoặc thiết bị mạng, nhà phát triển có thể quyết định sử dụng một giao thức mở hoặc tự mình xây dựng một giao thức độc quyền riêng.

Một số giao thức được mô tả trong RFC (Request for Comments, có địa chỉ https://www.rfc-editor.org/) và có thể truy cập tự do. Loại giao thức như vậy được gọi là giao thức mở.

Ví dụ, giao thức của ứng dụng web HTTP được mô tả trong RFC 2616. Nếu một nhà phát triển trình duyệt tuân thủ các quy tắc trong RFC của HTTP, trình duyệt đó có thể truy xuất trang web từ bất kỳ chương trình máy chủ web nào (với điều kiện chương trình máy chủ cũng phải tuân thủ RFC của HTTP).

Một số giao thức tầng ứng dụng khác không được mô tả trong RFC mà đi kèm với ứng dụng hoặc thiết bị cụ thể. Ngoại trừ nhà phát triển phần mềm đó, những người khác không biết gì về giao thức này, và do đó, không (hoặc khó) thực hiện được giao thức đó cho ứng dụng của mình. Các loại giao thức như vậy gọi là giao thức độc quyền. Ví dụ, Skype sử dụng một số giao thức độc quyền.

4.2.2. Dịch vụ truyền thông

Phần mềm thực thi giao thức được sử dụng để thực hiện các chức năng truyền thông tin. Các chức năng này có thể được khai thác bởi các chương trình ứng dụng hoặc các giao thức khác. Những chức năng truyền thông tin được khai thác bởi chương trình hoặc giao thức khác được gọi là **dịch vụ truyền thông**. Các dịch vụ truyền thông mạng liên quan đến các giao thức và mỗi giao thức cung cấp một hoặc một số dịch vụ truyền thông nhất định cho ứng dụng.

Các dịch vụ truyền thông có thể được phân loại theo nhiều cách khác nhau:

- Khi xem xét từ khía cạnh "ai gửi cho ai", có hai loại dịch vụ truyền thông cơ bản mà các giao thức cung cấp: truyền thông *end-to-end* và truyền thông *host-to-host.* Khi lập trình ứng dụng mạng, chỉ có thể sử dụng hai loại dịch vụ truyền thông end-to-end và host-to-host.

- Khi xem xét từ khía cạnh "chất lượng", có hai loại dịch vụ truyền thông: *truyền thông tin cậy* và *truyền thông không tin cậy.* Việc chọn loại dịch vụ nào phụ thuộc vào đặc điểm của từng loại ứng dụng mạng.

- Khi xem xét từ khía cạnh "quan hệ giữa các bên khi truyền dữ liệu", có dịch vụ *truyền thông hướng liên kết* và dịch vụ *truyền thông phi liên kết.*

- Khi xét theo bản chất truyền dữ liệu, có thể phân biệt giữa *truyền thông thực* (vật lý) và *truyền thông ảo* (logic).

a) Truyền thông end-to-end

Các dịch vụ truyền thông mà ứng dụng sử dụng để truyền dữ liệu của mình tới một ứng dụng chạy trên một thiết bị đầu cuối khác gọi là dịch vụ **truyền thông đầu cuối - đầu cuối** (end-to-end communications). Nói đơn giản, đây là loại truyền thông giữa các ứng dụng chạy trên các thiết bị đầu cuối khác nhau.

Đây là loại dịch vụ truyền thông phổ biến nhất mà phần mềm mạng sử dụng. Gần như tất cả các loại ứng dụng mạng đều phải sử dụng đến loại truyền thông này.

Loại dịch vụ truyền thông này thường do hệ điều hành cung cấp dưới dạng các hàm API hệ thống. Chương trình ứng dụng gọi tới các hàm này để yêu cầu mang dữ liệu của mình tới chương trình đối tác.

Do ứng dụng mạng có nhiều thành phần, để các thành phần này có thể “nói chuyện” với nhau, chúng phải đưa ra các thỏa thuận về cấu trúc dữ liệu, trình tự gửi nhận dữ liệu, và các thao tác phải thực hiện khi gửi/nhận dữ liệu. Nói cách khác, tự bản thân chương trình ứng dụng phải đưa ra giao thức của riêng mình và sử dụng dịch vụ truyền thông end-to-end để thực thi giao thức đó.

b) Truyền thông host-to host

Việc truyền dữ liệu từ thiết bị đầu cuối này tới thiết bị đầu cuối khác được gọi là **truyền thông host-to-host**.

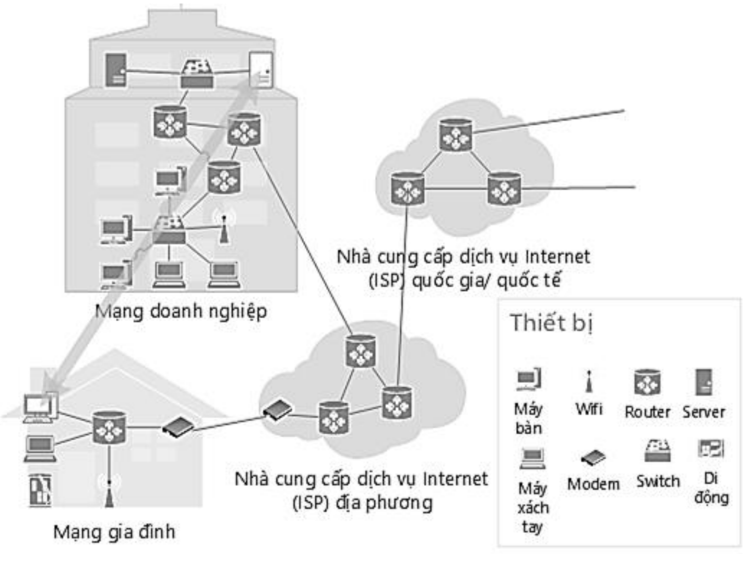
Truyền thông host-to-host không phân phối dữ liệu đến cho ứng dụng cụ thể nào, cũng không phục vụ cho ứng dụng cụ thể nào. Loại truyền thông này chỉ giúp dữ liệu (của nhiều ứng dụng khác nhau) đi từ thiết bị đầu cuối này tới thiết bị đầu cuối khác.

Dịch vụ truyền thông end-to-end phải sử dụng tới dịch vụ truyền host-to-host. Nói cách khác, dịch vụ truyền host-to-host là nền tảng hoạt động cho dịch vụ end-to-end. Ví dụ, khi gửi thư đến một căn hộ trong một tòa chung cư lớn, dịch vụ bưu chính chỉ chịu trách nhiệm đưa thư đến tòa chung cư (host-to-host); bản thân chung cư sẽ có dịch vụ phân phối thư đến từng căn hộ (end-to-end).

Để thực hiện nhiệm vụ truyền host-to-host, trên cả thiết bị đầu cuối và router đều phải cài đặt chung một giao thức (giao thức IP trong mạng TCP/IP mà chúng ta sẽ gặp ở phần sau của bài học này). Giao thức IP được thực thi ở dạng phần mềm hệ thống trên thiết bị đầu cuối (mà chúng ta có thể sử dụng), và là một phần mềm chạy trên hệ điều hành chuyên dụng của bộ định tuyến (router).

Lập trình mạng với socket trong giáo trình này chỉ có thể thực hiện được hai loại truyền thông end-to-end và host-to-host.

Cả truyền thông end-to-end và host-to-host đều là truyền thông ảo (virtual communication channel) với ý nghĩa: các host hoặc các ứng dụng "cảm giác" như đang được "nói chuyện" trực tiếp với nhau, mặc dù giữa chúng là rất nhiều loại môi trường truyền và các thiết bị mạng trung gian.


Mạng doanh nghiệp
Nhà cung cấp dịch vụ Internet (ISP) quốc gia/ quốc tế
Mạng gia đình
Nhà cung cấp dịch vụ Internet (ISP) địa phương
Thiết bị
Máy bàn
Wifi
Router
Server
Máy xách tay
Modem
Switch
Di động

Hình 4.3. Truyền thông vật lý và truyền thông ảo

c) Truyền thông tin cậy và truyền thông không tin cậy

Một số giao thức cung cấp dịch vụ truyền thông tin cậy. **Truyền thông tin cậy** là loại dịch vụ truyền thông trong đó giao thức đảm bảo rằng dữ liệu sẽ được truyền đi tới đích nguyên vẹn, không mất mát, không sai lệch, đúng thứ tự.

Ngược lại, **truyền thông không tin cậy** là loại dịch vụ truyền thông trong đó giao thức không đảm bảo rằng dữ liệu sẽ được truyền tới đích nguyên vẹn. Nghĩa là có thể xảy ra mất dữ liệu, sai dữ liệu, hoặc dữ liệu tới không đúng thứ tự.

Có thể đặt câu hỏi, tại sao phải có dịch vụ truyền thông "không tin cậy", vì rõ ràng dịch vụ truyền tin cậy có vẻ rất tốt. Vấn đề nằm ở chỗ, để đảm bảo cho truyền thông tin cậy, giao thức có thể cần thực hiện rất nhiều công việc khác nhau. Thông thường nhất là mỗi khi gửi đi một phần dữ liệu, máy nguồn phải chờ phản hồi từ máy đích. Các công việc này dẫn đến một kết quả là tốc độ truyền dữ liệu theo dịch vụ truyền tin cậy sẽ thấp. Ngược lại, truyền thông không tin cậy đơn giản hơn và tốc độ cao hơn rất nhiều.

d) Truyền thông hướng liên kết và truyền thông phi liên kết

Dịch vụ **truyền thông hướng liên kết** là loại dịch vụ truyền thông trong đó hai bên tham gia vào quá trình truyền phải trao đổi trước các thông tin để chuẩn bị cho quá trình truyền dữ liệu chính thức. Việc trao đổi thông tin trước khi truyền thường được gọi là quá trình thiết lập liên kết.

Sau khi thiết lập liên kết, hai bên tham gia truyền thông lưu lại thông tin về nhau và có thể bắt đầu quá trình trao đổi dữ liệu. Dữ liệu trao đổi qua liên kết thường không bị giới hạn về kích thước. Nếu kích thước dữ liệu quá lớn, nó có thể được cắt thành từng phần nhỏ và liên tục chuyển đi, tương tự như một băng chuyền trong nhà máy.

Sau khi quá trình trao đổi dữ liệu hoàn tất, liên kết có thể bị hủy bỏ. Khi hủy bỏ liên kết, tất cả thông tin về phiên làm việc vừa xong sẽ bị xóa bỏ. Hai bên không còn liên hệ gì. Nếu muốn trao đổi dữ liệu, liên kết cần thiết lập lại từ đầu.

Trong dịch vụ **truyền thông phi liên kết**, hai bên tham gia truyền thông không cần trao đổi thông tin trước khi truyền.

Khi sử dụng dịch vụ này, bên nguồn có thể gửi dữ liệu đi ngay lập tức mà không cần thực hiện bất kỳ thủ tục nào, còn bên đích không biết gì về bên nguồn cho đến khi nó nhận được dữ liệu.

Dữ liệu truyền theo dịch vụ này thường được đóng thành từng gói độc lập, có kích thước giới hạn, và được truyền riêng rẽ, tương tự như việc gửi bưu phẩm.

Do hai bên không lưu trữ thông tin về nhau, các bên có thể kết thúc quá trình truyền dữ liệu bất kỳ lúc nào mà không cần thực hiện thủ tục gì khác.

e) Truyền thông thực và truyền thông ảo

Truyền thông thực (hay truyền thông vật lý) là quá trình truyền tín hiệu điện, sóng vô tuyến hoặc ánh sáng qua các môi trường truyền dẫn vật lý như dây đồng, cáp quang, sóng radio... Giữa hai thiết bị trong mạng, dữ liệu được truyền đi qua một chuỗi các kết nối vật lý, bao gồm dây dẫn và các thiết bị trung gian khác. Quá trình này là hữu hình, đo đếm được, và liên quan đến phần cứng thực tế. Truyền thông thực là nền tảng vật lý cho mọi hoạt động truyền dữ liệu.

Truyền thông ảo (virtual communication) là khái niệm logic phản ánh cách các giao thức mô phỏng việc truyền dữ liệu giữa hai thực thể như thể chúng đang giao tiếp trực tiếp với nhau, dù thực tế có thể thông qua nhiều lớp trung gian. Ví dụ, một ứng dụng trên máy tính A gửi dữ liệu đến ứng dụng trên máy tính B qua giao thức TCP. Cả hai ứng dụng "cảm nhận" như đang trao đổi trực tiếp, dù dữ liệu thực tế đã đi qua nhiều tầng giao thức, thiết bị và môi trường vật lý.

Truyền thông ảo được hình thành nhờ các giao thức và cơ chế phân tầng, giúp trừu tượng hóa hệ thống mạng, hỗ trợ lập trình dễ dàng hơn, và ẩn đi sự phức tạp của lớp vật lý bên dưới. Truyền thông ảo là lớp trừu tượng hóa logic do phần mềm và giao thức tạo ra để đơn giản hóa việc giao tiếp giữa các thực thể trong mạng.

4.3. Bộ giao thức TCP/IP

4.3.1. Tổng quan về bộ giao thức TCP/IP

Trên thiết bị đầu cuối đồng thời có rất nhiều giao thức cùng hoạt động: những giao thức gắn với các phần mềm ứng dụng và thường được tích hợp trong các phần mềm này; những giao thức cung cấp dịch vụ truyền thông end-to-end và host-to-host do hệ điều hành cung cấp. Ngoài ra còn có những giao thức gắn với các công nghệ và thiết bị phần cứng.

Các giao thức này không hoạt động độc lập mà có sự quan hệ với nhau. Quan hệ giữa các giao thức này gần giống như cách gọi hàm / phương thức trong các ngôn ngữ lập trình. Theo đó, giao thức gắn với ứng dụng sẽ gọi tới giao thức end-to-end. Giao thức end-to-end lại gọi tới giao thức host-to-host.

Một nhóm các giao thức trên thiết bị đầu cuối có quan hệ với nhau như vậy tạo thành một **bộ giao thức**.

Hiện nay, **bộ giao thức TCP/IP** là bộ giao thức phổ biến nhất, là chuẩn công nghiệp được hỗ trợ bởi mọi nhà sản xuất phần cứng và phần mềm mạng.

Bộ giao thức TCP/IP bao gồm rất nhiều giao thức khác nhau và được chia vào các nhóm, mỗi nhóm được gọi là một **tầng**. Tổng cộng có 4 tầng (tức là 4 nhóm giao thức), mỗi tầng được đặt một tên gọi riêng, bao gồm: tầng ứng dụng, tầng giao vận, tầng liên mạng, tầng liên kết. Trong đó:

- **Tầng ứng dụng** (application layer) chứa các giao thức mà thực thi của nó tích hợp trong các phần mềm ứng dụng. Các giao thức tầng ứng dụng phổ biến cùng với phần mềm sử dụng chúng bao gồm:
    - HTTP (HyperText Transfer Protocol): sử dụng bởi trình duyệt và chương trình chủ web;
    - SMTP, POP3, IMAP: sử dụng bởi chương trình Email client và server;
    - FTP (File Transfer Protocol): sử dụng bởi chương trình FTP client và server.
- **Tầng giao vận** (transport layer) chứa các giao thức cung cấp dịch vụ truyền thông end-to-end, chủ yếu bao gồm:
    - Giao thức TCP (Transmission Control Protocol): cung cấp dịch vụ truyền thông *tin cậy hướng liên kết*;
    - UDP (User Datagram Protocol): cung cấp dịch vụ truyền thông *không tin cậy phi liên kết* tốc độ cao.
- **Tầng liên mạng** (internet layer) chứa các giao thức cung cấp dịch vụ truyền thông host-to-host:
    - Giao thức IP (Internet Protocol): cung cấp dịch vụ truyền thông host-to-host phi liên kết không tin cậy
    - Giao thức IPv6 (Internet Protocol v6): có vai trò tương tự như giao thức IP.
- **Tầng liên kết** (link layer) chứa các giao thức giúp kết nối thiết bị đầu cuối với mạng cục bộ:
    - Giao thức ARP (Address Resolution Protocol): dùng để xác định địa chỉ vật lý MAC gắn với địa chỉ logic IP của thiết bị.
    - Giao thức RARP (Reverse ARP): dùng để xác định địa chỉ logic gắn với địa chỉ vật lý của thiết bị (giao thức đã cũ).
    - Giao thức NDP (Neighbor Discovery Protocol): phiên bản ARP cho IPv6.
    - Giao thức DHCP (Dynamic Host Configuration Protocol) và DHCPv6: tự động gán thông số cấu hình cho thiết bị.

Mặc dù ở trên đã trình bày 4 tầng TCP/IP với rất nhiều giao thức, thực tế không phải tất cả các giao thức đó đều phải thực thi trên các thiết bị đầu cuối.

Tối thiểu, bộ giao thức TCP/IP yêu cầu phải thực thi các giao thức sau:

- Internet Protocol (IP),
- Address Resolution Protocol (ARP),
- Internet Control Message Protocol (ICMP),

- Transmission Control Protocol (TCP),
- User Datagram Protocol (UDP),
- Internet Group Management Protocol (IGMP).

Ngoài ra, giao thức Internet Protocol version 6 yêu cầu thêm Neighbor Discovery Protocol (NDP), ICMPv6, và Multicast Listener Discovery (MLD).

Có thể thấy rằng, đây đều là các giao thức ở tầng 2 và 3. Tất cả các giao thức ở tầng ứng dụng đều không bắt buộc vì chúng đi kèm với ứng dụng.

Các thiết bị mạng thông thường không hỗ trợ tất cả các giao thức trong bộ TCP/IP. Tùy thuộc vào loại thiết bị, thường chỉ có một bộ phận của bộ giao thức này được hỗ trợ. Do vậy, cũng có thể xác định vai trò của thiết bị đó trong mạng thông qua những giao thức TCP/IP được thiết bị đó hỗ trợ.

4.3.2. Giao thức UDP

Giao thức UDP (User Datagram Protocol) được định nghĩa trong RFC 768. Đây là một trong hai giao thức của tầng giao vận trong bộ giao thức TCP/IP và cũng là một trong những giao thức bắt buộc của TCP/IP.

a) Đặc điểm của UDP

Giao thức UDP được thực thi ở dạng phần mềm và cài đặt ở dạng thư viện hệ thống của hầu hết các hệ điều hành hiện nay.

UDP cung cấp dịch vụ vận chuyển dữ liệu đầu cuối - đầu cuối (end-to-end) cho chương trình ứng dụng. Phần mềm ứng dụng có thể trực tiếp sử dụng dịch vụ của UDP thông qua Socket API.

UDP là một giao thức đơn giản cung cấp dịch vụ phi liên kết (connectionless service) cho ứng dụng. Khi sử dụng UDP không tạo ra kết nối giữa các chương trình tham gia truyền thông.

UDP cung cấp dịch vụ truyền dữ liệu không tin cậy (unreliable data transfer service). Khi một chương trình gửi dữ liệu qua UDP, giao thức UDP không đảm bảo dữ liệu đó sẽ đến đích, đồng thời, các dữ liệu truyền tới đích có thể không theo đúng thứ tự như khi nó được gửi đi. UDP không có cơ chế kiểm soát nghẽn, do đó máy nguồn có thể gửi dữ liệu theo bất kỳ tốc độ nào.

Giao thức UDP thường được sử dụng trong các ứng dụng thời gian thực do các ứng dụng loại này có thể chấp nhận thất thoát dữ liệu nhưng yêu cầu tốc độ truyền dữ liệu cao hơn. Do tường lửa trên các thiết bị luôn có xu hướng chặn phần lớn các loại dòng dữ liệu UDP nên cần chú ý khi lựa chọn giao thức UDP.

b) Đóng gói dữ liệu UDP

Khi chúng ta gọi hàm UDP để vận chuyển một chuỗi byte tới tiến trình đích, chương trình UDP sẽ bổ sung thêm một chuỗi byte nhỏ (8 byte) vào đầu của chuỗi byte dữ liệu. Chuỗi byte bổ sung của UDP được gọi là header. Phần dữ liệu UDP nhận từ ứng dụng được gọi là phần payload.

Offset		Octet#	0								1								2								3							
Octet	Bit	Bit#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0		Source port																Destination port															
4	32		Length																Checksum															

Hình 4.4. Cấu trúc header của gói tin UDP

UDP header cùng với payload tạo ra một đơn vị thông tin riêng gọi là datagram. Như vậy có thể dễ dàng hình dung, datagram thực chất cũng chỉ là một chuỗi byte lớn. Cấu trúc của UDP Header được trình bày ở Hình 4.4.

Ý nghĩa các trường trong UDP header như sau:

- Source port number (2 byte) là số cổng của tiến trình nguồn.
- Destination port number (2 byte) là số cổng tiến trình đích.
- Do trường Source port và Destination port chỉ sử dụng 2 byte (16 bit) để chứa giá trị, giá trị của nó phải nằm trong dải [0, 65535], tức là từ 0 đến 2^16-1.
- Length là độ dài của Header + data. Về lý thuyết thì 8 <= Length <= 65 535 (tức là 8 byte header + 65527 byte data), nhưng trên thực tế, 8 <= Length <= 65,507 bytes (bằng giá trị tối đa mô tả được bằng 2 byte là 65535 trừ đi 8 byte UDP header, trừ tiếp 20 byte IP header).
- Checksum dùng để kiểm tra lỗi header và dữ liệu nhằm đảm bảo tính toàn vẹn của gói tin. Checksum không bắt buộc với IPv4 (nếu không dùng thì chứa toàn các bit 0) nhưng bắt buộc với IPv6. Trường này được UDP tính toán tự động theo thuật toán mô tả trong RFC.

c) Truyền dữ liệu với giao thức UDP

Truyền tải dữ liệu qua UDP không tạo liên kết ảo giữa các tiến trình tham gia truyền thông. Do không tạo ra liên kết ảo trước khi truyền dữ liệu, giao thức UDP có thể truyền dữ liệu đi ngay lập tức mà không cần thực hiện quá trình xây dựng liên kết phức tạp. Vì lý do này mà chỉ khi nào nhận được dữ liệu, tiến trình mới biết được nó đang trao đổi dữ liệu với tiến trình nào.

Giao thức UDP đơn giản và cho phép truyền dữ liệu theo thời gian thực. Tuy nhiên, do thiếu các cơ chế kiểm soát dữ liệu trong quá trình truyền, giao thức UDP chỉ thích hợp khi truyền các loại dữ liệu không có yêu cầu cao về kiểm tra và sửa lỗi. Vì vậy UDP được sử dụng phổ biến nhất trong việc truyền tải dữ liệu đa phương tiện (video, audio, image), trong đó, việc mất một số datagram không ảnh hưởng đến việc sử dụng dữ liệu, đồng thời cần tốc độ truyền tải cao.

Truyền tải dữ liệu qua UDP không sử dụng các bộ nhớ đệm, do đó chương trình đích bắt buộc phải nhận gói tin UDP kịp thời. Nếu không phát lệnh nhận kịp thời, gói tin UDP sẽ mất đi.

Do giới hạn về kích thước tối đa của mỗi datagram, chương trình ứng dụng phải tự mình quản lý việc cắt và ghép dữ liệu nếu cần truyền dữ liệu có kích thước lớn qua UDP.

Giao thức và dịch vụ tầng ứng dụng phổ biến sử dụng UDP bao gồm: Domain Name System (DNS), Network Time Protocol (NTP), IP tunneling, Remote Procedure Call (RPC), Network File System (NFS), DHCP, TFTP (Trivial FTP), IPTV (Streaming media).

4.3.3. Giao thức TCP

TCP được định nghĩa trong RFC 793. TCP là một trong hai giao thức tầng giao vận của TCP/IP và là một trong những giao thức bắt buộc của bộ giao thức TCP/IP. Bản thân tên gọi của bộ giao thức này xuất phát từ chính tên của giao thức TCP.

a) Đặc điểm của TCP

Giao thức TCP được thực thi ở dạng phần mềm và cài đặt ở dạng thư viện hệ thống của hầu hết các hệ điều hành hiện nay.

Giao thức TCP cung cấp dịch vụ truyền thông end-to-end trực tiếp cho ứng dụng thông qua TCP socket API. Các ứng dụng có thể trực tiếp gọi tới TCP socket để yêu cầu truyền dữ liệu.

Giao thức TCP cung cấp cho ứng dụng dịch vụ truyền thông end-to-end hướng kết nối (connection-oriented service) và tin cậy (reliable data transfer service):

- **Dịch vụ hướng kết nối**: trong truyền thông qua giao thức TCP, các chương trình phải trao đổi thông tin kiểm soát trước khi dữ liệu của ứng dụng được truyền đi. Thủ tục này được gọi là ‘bắt tay ba bước’ và tạo ra một kết nối TCP (TCP connection) giữa hai chương trình. Đây là loại kết nối song công (full-duplex), nghĩa là hai chương trình có thể đồng thời gửi dữ liệu cho nhau. Khi ứng dụng hoàn thành việc gửi dữ liệu, kết nối TCP có thể được hủy bỏ.
- **Dịch vụ truyền tin cậy**: TCP đảm bảo các chương trình tham gia truyền thông có thể truyền toàn bộ dữ liệu mà không bị lỗi và tới theo đúng thứ tự. Nói một cách khác, khi một chương trình nguồn truyền dữ liệu đi, TCP đảm bảo chương trình đích sẽ nhận được đúng dữ liệu đó, không mất hoặc dư thừa byte nào.

TCP được sử dụng nếu đặt ra yêu cầu cao về tính toàn vẹn của dữ liệu khi truyền nhưng có thể chấp nhận tốc độ truyền thấp.

b) Đóng gói dữ liệu

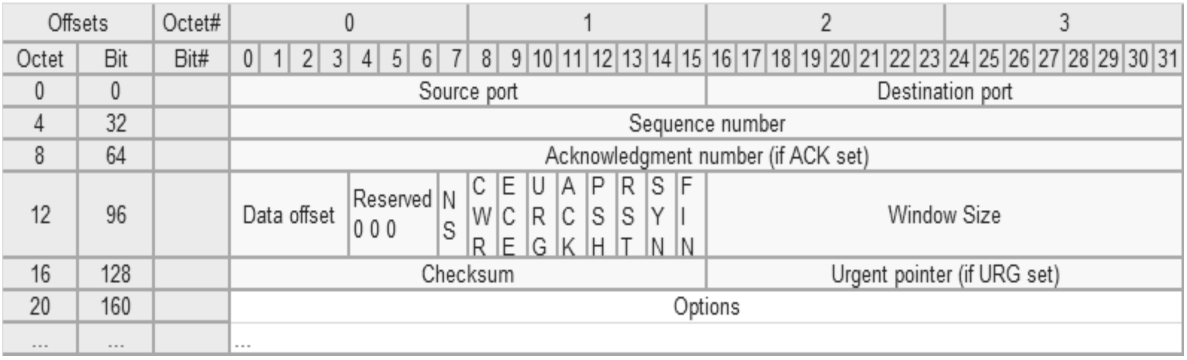
TCP chỉ vận chuyển các chuỗi byte (mà ứng dụng tạo ra) mà không quan tâm tới ý nghĩa của chúng.

Dữ liệu của ứng dụng sau khi chuyển qua socket tới tầng giao vận sẽ được TCP tách thành từng phần. Quá trình này được gọi là phân mảnh dữ liệu (segmentation).

Đối với mỗi phần dữ liệu này, TCP sẽ bổ sung thêm các thông tin điều khiển vào trước phần dữ liệu để tạo thành một đơn vị dữ liệu của tầng giao vận, gọi là TCP segment.

Phần thông tin điều khiển mà TCP thêm vào trước khối dữ liệu ứng dụng gọi là TCP header. Phần dữ liệu ứng dụng lưu trong TCP segment được gọi là payload.

Cấu trúc của TCP header được trình bày trong Hình 4.5.



<table>
<tr><td colspan="2">Offsets</td><td>Octet#</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Octet</td><td>Bit</td><td>Bit#</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td>0</td><td></td><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><td>4</td><td>32</td><td></td><td colspan="32">Sequence number</td></tr>
<tr><td>8</td><td>64</td><td></td><td colspan="32">Acknowledgment number (if ACK set)</td></tr>
<tr><td>12</td><td>96</td><td></td><td colspan="4">Data offset</td><td colspan="3">Reserved
0 0 0</td><td>NS</td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Window Size</td></tr>
<tr><td>16</td><td>128</td><td></td><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><td>20</td><td>160</td><td></td><td colspan="32">Options</td></tr>
<tr><td>...</td><td>...</td><td></td><td colspan="32">...</td></tr>
</table>

Hình 4.5. Cấu trúc TCP header

Ý nghĩa các trường của TCP header được đưa vào dưới đây cho đầy đủ và mang ý nghĩa tham khảo. Sinh viên không cần thiết phải ghi nhớ hết các trường này. Các trường quan trọng hơn cả cần nhớ là Source Port, Destination Port, Seq Number, ACK Number, bit SYN và bit FIN.

Ý nghĩa các trường như sau:

- Source port (16 bit): số cổng của tiến trình gửi.
- Destination port (16 bit): số cổng của của tiến trình nhận.
- Sequence number (32 bit): số thứ tự. Giá trị và ý nghĩa của số thứ tự phụ thuộc vào giá trị của bit SYN.
- Acknowledgement number (32 bit): số báo nhận. Trường số thứ tự và số báo nhận được sử dụng để đảm bảo dịch vụ truyền dữ liệu tin cậy.
- Các trường flag (6 trường, mỗi trường 1 bit):
    - Bit ACK dùng để xác nhận tính chính xác của giá trị trường Acknowledgement.
    - Bit RST, SYN, FIN dùng để thiết lập và hủy bỏ kết nối.
    - Bit PSH chỉ định máy nhận phải chuyển dữ liệu lên tầng trên ngay lập tức.
    - Bit URG chỉ định rằng có vùng dữ liệu được đánh dấu là “urgent” (dữ liệu khẩn). Nếu URG=1, trường Urgent pointer (16 bit) sẽ chứa vị trí của byte cuối cùng của dữ liệu khẩn; TCP phải thông báo về sự tồn tại của dữ liệu khẩn cho

máy nhận, đồng thời chuyển cho máy nhận một con trỏ tới cuối của vùng dữ liệu khẩn. Trong thực tế, hai trường PSH và URG không được sử dụng.

- Checksum (16 bit): sử dụng để kiểm tra lỗi.
- Window size (16 bit): kích thước cửa sổ dùng trong kiểm soát luồng.
- Reserved (3 bit): luôn chứa giá trị 000, để dành sử dụng trong tương lai.
- Data offset (4 bit), còn gọi là Header length: kích thước của TCP header. Kích thước của TCP header có thể thay đổi do sự tồn tại của trường Options. Tuy nhiên, trường Options này thường để trống, và do đó, kích thước của TCP header thường là 20 byte.
- Options (độ dài thay đổi): được sử dụng khi hai máy tham gia truyền thông thỏa thuận về kích thước tối đa của segment (TA: maximum segment size, MSS) hoặc hệ số tỉ lệ của cửa sổ trượt (dùng cho các mạng tốc độ cao).

Quy trình hoạt động của TCP chia làm 3 giai đoạn: Thiết lập liên kết, Truyền dữ liệu và Đóng liên kết.

c) Thiết lập liên kết TCP

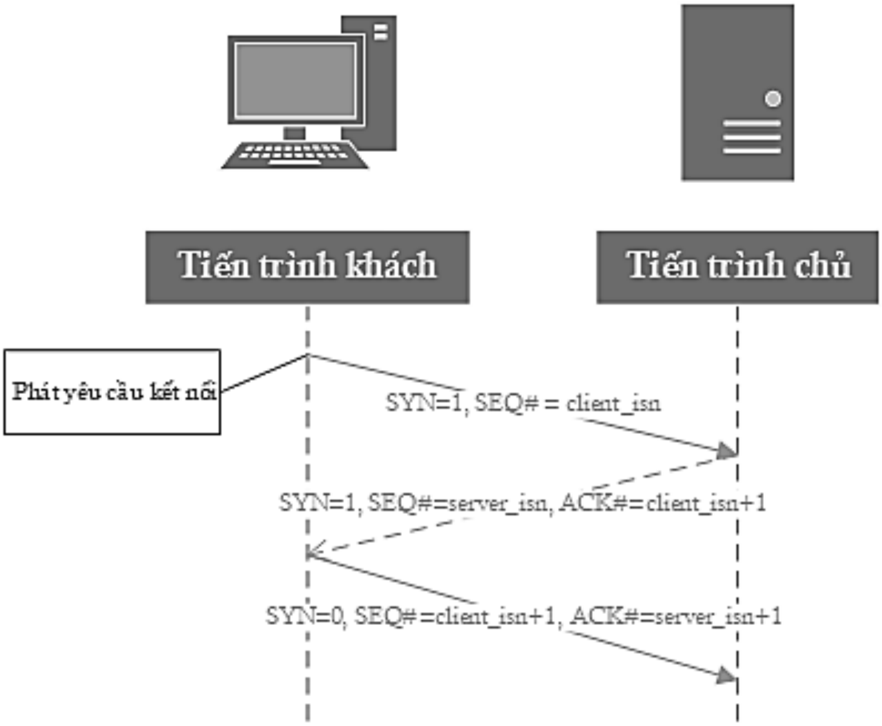
Khi một tiến trình muốn tiến hành truyền thông với với một tiến trình khác qua mạng sử dụng giao thức TCP, nó phải thông báo cho tầng giao vận để khởi tạo một liên kết TCP. Trong quá trình này, tiến trình khởi tạo liên kết được gọi là tiến trình khách, tiến trình còn lại tiếp nhận liên kết TCP được gọi là tiến trình chủ.

Quá trình thiết lập liên kết TCP giữa hai tiến trình được thực hiện trong ba bước và thường được gọi là quá trình **bắt tay ba bước** (three-way handshake).

Bước 1. Tiến trình khách gửi một TCP segment không chứa dữ liệu tới tiến trình chủ, trong đó, bit SYN = 1. Tiến trình khách sẽ chọn một giá trị ngẫu nhiên cho trường Sequence (để tránh tấn công loại SYN). Segment loại này thường được gọi là SYN segment.

Bước 2. Khi tiến trình chủ nhận được SYN segment, nó sẽ bố trí bộ nhớ cho TCP buffer và các biến trạng thái cho liên kết TCP. Tiến trình chủ sau đó sẽ gửi một segment trả lời cho tiến trình khách, trong segment này: bit SYN = 1; số báo nhận (ACK number) = Sequence của SYN segment (nhận từ tiến trình khách) + 1; số thứ tự Sequence được lựa chọn ngẫu nhiên. Segment trả lời này thường được gọi là SYNACK segment.

Bước 3. Khi nhận được SYNACK segment, tiến trình khách cũng thiết lập bộ nhớ đệm và các biến trạng thái cho liên kết TCP, sau đó gửi lại cho tiến trình chủ một segment, trong đó, trường số báo nhận ACK = số thứ tự Sequence của server + 1, SYN = 0, trường số thứ tự Sequence tăng thêm 1 đơn vị.


Tiến trình khách
Tiến trình chủ
Phát yêu cầu kết nối
SYN=1, SEQ# = client_isn
SYN=1, SEQ#=server_isn, ACK#=client_isn+1
SYN=0, SEQ#=client_isn+1, ACK#=server_isn+1

Hình 4.6. Quá trình bắt tay ba bước của TCP

Sau 3 bước này, liên kết TCP giữa hai tiến trình đã được thiết lập.

Liên kết TCP là loại liên kết song công (TA: full-duplex). Tức là, một khi thiết lập được liên kết giữa tiến trình A và tiến trình B, dữ liệu tầng ứng dụng có thể đồng thời truyền từ A tới B và từ B tới A.

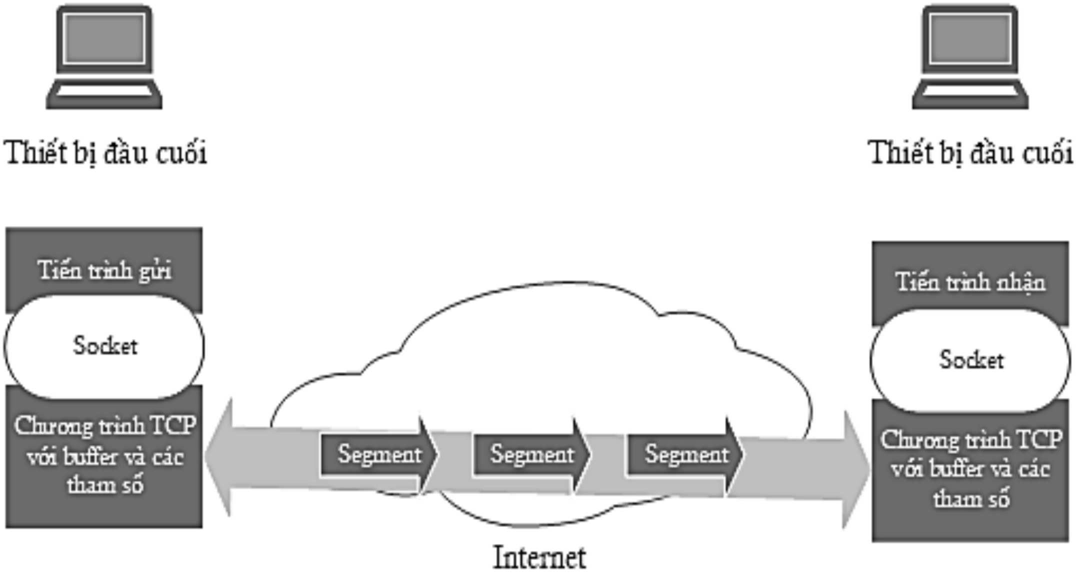
Liên kết TCP thuộc loại liên kết điểm-điểm (TA: point-to-point) giữa duy nhất một tiến trình gửi và một tiến trình nhận. Trong TCP không tồn tại loại truyền thông kiểu quảng bá (TA: multicasting) giữa một tiến trình gửi với nhiều tiến trình nhận.

d) Truyền dữ liệu

Khi đã hình thành liên kết TCP, hai tiến trình tham gia liên kết có thể truyền dữ liệu cho nhau. Khi một tiến trình chuyển một chuỗi byte qua socket tới tầng giao vận, dữ liệu đó sẽ hoàn toàn do TCP trên máy đó quản lý và chịu trách nhiệm.

TCP không lập tức đóng gói dữ liệu và chuyển lên mạng. Thay vào đó, TCP chuyển dữ liệu tới một bộ nhớ đệm (buffer) được hình thành trong quá trình bắt tay ba bước. Theo thời gian, TCP sẽ lấy dữ liệu (dữ liệu của ứng dụng) từ bộ nhớ đệm, đóng gói thành segment, và chuyển lên đường truyền.

Trong khoảng thời gian chờ ở bộ nhớ đệm, ứng dụng có thể gửi thêm dữ liệu và dữ liệu này sẽ tiếp tục được bổ sung vào bộ nhớ đệm. Tương tự, TCP trên máy đích cũng sử dụng một bộ nhớ đệm cục bộ của riêng mình để lưu dữ liệu nhận được cho đến khi nó nhận được đầy đủ tất cả các gói tin theo đúng thứ tự. Sau đó, dữ liệu này mới được chuyển lên cho ứng dụng.


Thiết bị đầu cuối
Thiết bị đầu cuối
Tiến trình gửi
Socket
Chương trình TCP với buffer và các tham số
Segment
Segment
Segment
Tiến trình nhận
Socket
Chương trình TCP với buffer và các tham số
Internet

Hình 4.7. Quá trình truyền dữ liệu trong TCP

Đơn vị dữ liệu của TCP được gọi là segment. Kích thước tối đa của segment (Maximum Segment Size, MSS) được thiết lập tự động bằng cách xác định kích thước tối đa của frame dữ liệu của tầng liên kết. Việc thiết lập MSS như vậy đảm bảo cho TCP segment có thể nằm vừa khít trong một frame với kích thước đủ lớn để có hiệu suất gửi dữ liệu cao nhất.

Khi TCP gửi đi dữ liệu kích thước lớn (ví dụ, một file ảnh lớn trong một trang web), dữ liệu này sẽ được cắt thành từng phần có kích thước bằng MSS (trừ mảnh cuối cùng có thể có kích thước nhỏ hơn).

Ngược lại, các ứng dụng tương tác thường dùng các khối dữ liệu có kích thước nhỏ hơn MSS. Ví dụ, chương trình Telnet thường gửi đi dữ liệu có kích thước đúng 1 byte, do đó kích thước TCP segment dùng với Telnet thường chỉ có 21 byte.

Mỗi khi TCP trên máy nguồn gửi đi một segment, nó sẽ chờ nhận được một gói tin phản hồi (ACK) từ máy đối tác thông báo về tình trạng của segment vừa nhận. Chỉ khi có ACK xác nhận đạt yêu cầu, TCP trên máy nguồn mới tiếp tục gửi đi segment tiếp theo.

Do những đặc điểm trên, khi tiến trình gửi dữ liệu đi (dưới dạng các thông điệp thuộc tầng ứng dụng), TCP đảm bảo việc chuyển được toàn bộ dữ liệu tới máy đích theo đúng thứ tự, không mất mát và không lỗi.

Nhìn từ khía cạnh ứng dụng, liên kết và truyền dữ liệu qua TCP giống như một luồng byte liên tục "chảy" từ buffer của máy nguồn tới buffer của máy đích. Khi tới buffer của máy đích, dữ liệu sẽ chờ được lấy vào chương trình và xử lý.

e) Đóng liên kết

Cả hai tiến trình tham gia vào liên kết TCP đều có thể yêu cầu đóng liên kết.

Khi một trong hai tiến trình muốn kết thúc một liên kết, nó sẽ gửi đi một segment không chứa dữ liệu, trong đó bit FIN = 1.

Khi tiến trình còn lại nhận được segment này, nó sẽ gửi lại một ACK segment. Tiếp theo tiến trình này sẽ gửi tiếp một segment, trong đó bit FIN = 1.

Cuối cùng, tiến trình muốn kết thúc liên kết sẽ gửi một ACK segment. Ngay sau đó, bộ nhớ đệm và các biến trạng thái liên quan tới liên kết sẽ được giải phóng và liên kết được ngắt.

4.3.4. Giao thức IPv4

Giao thức IP (Internet Protocol) là một trong những giao thức quan trọng nhất mạng máy tính, là nòng cốt của mạng Internet, và là một giao thức bắt buộc trong bộ giao thức TCP/IP. Bản thân tên gọi bộ giao thức TCP xuất phát từ hai giao thức cơ bản quan trọng nhất của nó là TCP và IP.

Hiện nay có hai phiên bản phổ biến của IP được sử dụng là IPv4 và IPv6. IPv4 được đặc tả trong RFC 791, còn IPv6 được mô tả trong RFC 2460. Từ 2006, IPv6 bắt đầu được triển khai trên diện rộng. Đến nay, IPv6 đang dần thay thế cho IPv4. Tuy nhiên, IPv4 vẫn được sử dụng rất rộng rãi.

Do giới hạn về thời lượng giảng dạy, trong bài học này chỉ trình bày các nội dung liên quan đến IPv4. Tiếp theo đây, khi nói về giao thức IP hoặc địa chỉ IP, chúng ta sẽ hiểu đó là IPv4.

a) Đặc điểm của IP

Giao thức IP được cài đặt trên tất cả các thiết bị đầu cuối dưới dạng một dịch vụ của hệ thống. Giao thức IP cũng được cài đặt trên các thiết bị mạng tầng 3 (như router).

Giao thức IP cung cấp dịch vụ truyền thông host-to-host cho các giao thức khác, bao gồm các giao thức tầng giao vận (TCP và UDP) cũng như một số giao thức đặc biệt khác (như ICMP). Các chương trình ứng dụng không trực tiếp khai thác dịch vụ của IP. Tuy nhiên, các phần mềm ứng dụng có thể khai thác dịch vụ của IP thông qua IP socket API.

IP hoạt động theo nguyên tắc "best-effort". Theo đó,

(1) IP không cam kết một độ trễ nhất định cho việc truyền dữ liệu;

(2) IP không cung cấp cơ chế đảm bảo chất lượng dịch vụ nội tại;

(3) IP không bảo đảm sự ổn định của dịch vụ;

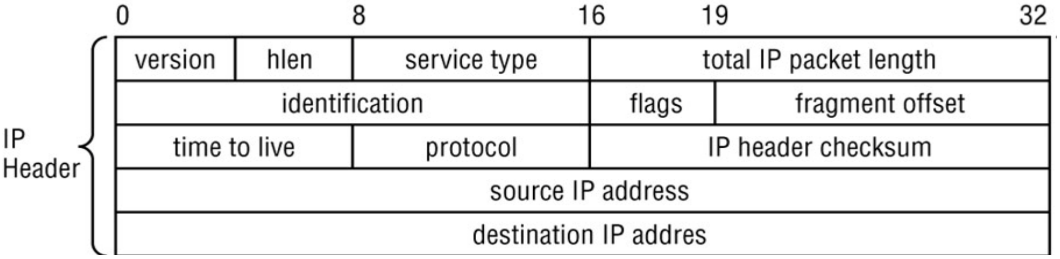
(4) IP không đảm bảo ưu tiên truyền thông cho các dịch vụ hoặc ứng dụng cụ thể;

(5) một gói dữ liệu có thể được truyền đi qua mạng ngay cả khi nó không hợp lệ hoặc không thể đến được đích.

Đặc điểm "best-effort" của giao thức IP phù hợp cho môi trường Internet bởi vì Internet là một mạng phức tạp với nhiều nguồn lưu lượng khác nhau và các loại dịch vụ. Mô hình này cho phép mạng linh hoạt và có khả năng mở rộng, nhưng đồng thời cũng đòi hỏi các ứng dụng phải tự xử lý các vấn đề liên quan đến độ trễ, độ tin cậy và ưu tiên truyền thông nếu cần.

b) Đóng gói dữ liệu với IP

PDU của giao thức IP có tên gọi là packet (gói tin). Mỗi gói tin IP là một chuỗi byte tạo ra từ PCI (header) có cấu trúc xác định và phần SDU là dữ liệu cần chuyển đi. PCI (hay header) của IP có cấu trúc như sau.



Hình 4.8. Cấu trúc header của IPv4

Ý nghĩa các trường của IP header được đưa vào đây cho đầy đủ và mang ý nghĩa tham khảo. Sinh viên không cần thiết phải ghi nhớ hết các trường này. Các trường quan trọng hơn cả cần nhớ là Source IP Address, Destination IP Address, Protocol.

Dưới đây là các thành phần/các trường của phần đầu gói tin IP:

- Version (4 bit): Xác định phiên bản của giao thức IP. Với IPv4, giá trị này là 4 (tức 0100 trong nhị phân).
- IHL - Internet Header Length (4 bit): Cho biết độ dài của header IP, tính bằng số từ 32 bit. Giá trị tối thiểu là 5 (tức 20 byte, nếu không có tuỳ chọn mở rộng).
- Type of Service - ToS (8 bit): Ưu tiên gói tin và yêu cầu chất lượng dịch vụ (QoS). Trường này cho phép phân biệt mức ưu tiên giữa các loại dữ liệu như thoại, video hay truyền tệp.
- Total Length (16 bit): Tổng độ dài của toàn bộ gói tin IP, bao gồm phần header và phần dữ liệu (payload). Giá trị tối đa là 65.535 byte.
- Identification (16 bit): Số định danh dùng để xác định các gói tin thuộc cùng một luồng truyền khi cần phân mảnh.
- Flags (3 bit): Điều khiển việc phân mảnh. Trong đó có các bit như:
- DF (Don't Fragment): không được phân mảnh gói tin.
- MF (More Fragments): cho biết còn phần mảnh phía sau.
- Fragment Offset (13 bit): Vị trí của phần mảnh hiện tại trong gói tin ban đầu, tính theo đơn vị 8 byte.

- TTL - Time To Live (8 bit): Giới hạn số lần gói tin có thể đi qua router. Mỗi lần đi qua một thiết bị định tuyến, TTL giảm 1. Nếu TTL = 0, gói bị loại bỏ để tránh vòng lặp mạng.
- Protocol (8 bit): Chỉ ra giao thức tầng trên sử dụng gói tin IP, ví dụ: TCP - 6, UDP - 17, ICMP - 1.
- Header Checksum (16 bit): Mã kiểm tra lỗi phần header IP. Nếu checksum sai, gói tin bị loại bỏ.
- Source Address (32 bit): Địa chỉ IP của thiết bị gửi.
- Destination Address (32 bit): Địa chỉ IP của thiết bị nhận.
- Options (tuỳ chọn - độ dài biến đổi): Dùng cho các chức năng mở rộng như định tuyến theo đường đi cụ thể, đánh dấu thời gian, bảo mật... Thường không được sử dụng trong các gói tin thông thường.
- Data (payload): Phần dữ liệu mang theo từ tầng giao vận (TCP/UDP) hoặc giao thức tầng trên.

4.3.5. Số cổng và địa chỉ IP

a) Số cổng - địa chỉ của tiến trình

Trên mỗi thiết bị đầu cuối đồng thời có rất nhiều tiến trình đồng thời khai thác các dịch vụ truyền thông do TCP hoặc UDP cung cấp. Khi dữ liệu đến đích TCP hoặc UDP cần một cơ chế để có thể phân biệt dữ liệu đó dành cho tiến trình nào.

Để TCP hoặc UDP xác định được tiến trình nào đang khai thác dịch vụ của mình, mỗi tiến trình được đánh một số không trùng lặp, gọi là *số cổng* (port).

Số cổng là loại địa chỉ dành cho các ứng dụng khai thác dịch vụ truyền thông end-to-end do TCP hoặc UDP cung cấp. Nói cách khác, đây là loại địa chỉ được sử dụng bởi giao thức TCP hoặc UDP để phân biệt các chương trình ứng dụng đang khai thác các dịch vụ truyền thông của hai giao thức này cung cấp.

Một số cổng của một dịch vụ thì không thể cấp cho 2 tiến trình. Ví dụ, nếu một tiến trình đã chiếm cổng TCP 80 thì tiến trình tiếp theo yêu cầu sử dụng cổng TCP này sẽ bị từ chối.

Số cổng cho dịch vụ TCP độc lập với số cổng cho dịch vụ UDP. Ví dụ, một tiến trình có thể chiếm dụng cổng TCP 80, đồng thời có thể chiếm dụng luôn cổng UDP 80. Một tiến trình đang chiếm cổng UDP 21 thì một tiến trình khác vẫn có thể chiếm dụng cổng TCP 21.

Số cổng là một số nguyên 2 byte (16 bit) với giá trị từ 0 đến 65535. Các tiến trình có thể chọn ngẫu nhiên bất kỳ giá trị số cổng nào, miễn là giá trị đó chưa được sử dụng cho tiến trình khác. Hệ điều hành cũng có thể chọn ngẫu nhiên số cổng để gán cho tiến trình.

Các tiến trình tham gia vào quá trình truyền thông phải thỏa thuận được với nhau về giá trị cổng. Các giá trị cổng nguồn (tiến trình gửi) và cổng đích (tiến trình nhận) đều được gửi đi kèm gói tin TCP (hoặc UDP) để các máy có thể xác định được cần chuyển gói tin đến cho tiến trình nào.

Mặc dù giá trị số cổng có thể gán ngẫu nhiên cho các tiến trình, một số ứng dụng mạng luôn chiếm một cổng cố định. Ví dụ, chương trình máy chủ web khi chạy luôn chiếm cổng 80, chương trình máy chủ FTP chiếm cổng 20, 21. Đây là những ứng dụng mạng phổ biến đi kèm với giao thức tầng ứng dụng của riêng chúng.

Giá trị cổng mà các chương trình này chiếm dụng được quy định trong các bảng đặc tả của giao thức. Các số cổng từ 0 đến 1023 thường được dùng cho các dịch vụ hệ thống hoặc các chương trình ứng dụng mạng phổ biến và thường được gọi là “cổng nổi tiếng” hoặc “cổng hệ thống”.

Dưới đây là danh sách một số cổng được “chiếm dụng” sẵn bởi các ứng dụng / giao thức mạng phổ biến:

Bảng 4.1. Các cổng nổi tiếng với dịch vụ sử dụng chúng

Cổng (Port)	Giao thức / Dịch vụ
20	FTP - Kênh dữ liệu
21	FTP - Kênh điều khiển
22	SSH Remote Login Protocol
23	Telnet
25	Simple Mail Transfer Protocol (SMTP)
53	Domain Name System (DNS)
80	HTTP
110	POP3
161	SNMP
443	HTTPS
546	DHCP Client
547	DHCP Server

b) Địa chỉ ip - địa chỉ logic của thiết bị đầu cuối

Trên một mạng máy tính thường đồng thời có rất nhiều thiết bị đầu cuối cùng hoạt động. Mỗi thiết bị đầu cuối đều được gán cho một địa chỉ duy nhất để phân biệt các thiết bị với nhau. Loại địa chỉ này có tên gọi là *địa chỉ IP*. Địa chỉ IP là loại địa chỉ giúp các thiết bị đầu cuối trao đổi dữ liệu với nhau trong liên mạng.

Tên gọi địa chỉ IP xuất phát từ giao thức IP (Internet Protocol) - giao thức sử dụng loại địa chỉ này để thực hiện truyền thông host-to-host xuyên qua liên mạng.

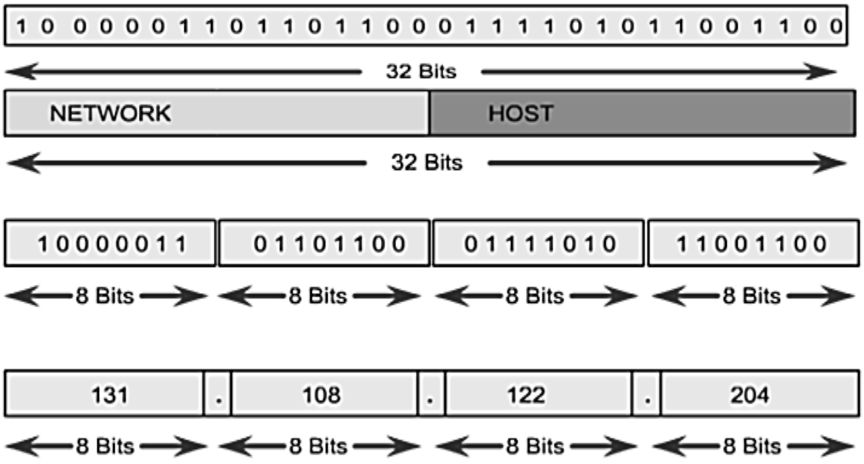
Khi nói về khía cạnh truyền thông tin, thiết bị đầu cuối cũng thường được gọi là **host**.

Địa chỉ IP, còn gọi là địa chỉ IPv4, là một số nguyên 4 byte (32 bit). Giá trị của địa chỉ IP do đó nằm trong dải từ 0 đến 2^32-1.

Để tiện lợi cho người dùng, số nguyên này được viết ở dạng mảng 4 byte. Mỗi byte được gọi là một octet và được biểu diễn bằng giá trị thập phân nằm trong khoảng từ 0 đến 255. Các octet được viết tách nhau bằng dấu chấm.

Ví dụ, địa chỉ IP thường gặp trong mạng LAN gia đình là 192.168.1.1.

Địa chỉ IPv4 ở dạng nhị phân được chia thành 2 phần: Phần **định danh mạng** (NETWORK) và **phần định danh host** (HOST). Định danh mạng là các bit liên tục ở bên trái. Định dang host là các bit còn lại ở bên phải.



Hình 4.9. Cấu trúc địa chỉ IPv4

Tất cả địa chỉ IP có chung phần định danh mạng được gọi là các *địa chỉ cùng dải*. Trong một mạng LAN, các host phải có địa chỉ IP cùng dải thì mới truyền được dữ liệu cho nhau.

Để chỉ rõ bao nhiêu bit dành cho phần network và bao nhiêu bit dành cho phần host, người ta thường thêm kí hiệu `/x` vào sau địa chỉ IP. Ví dụ, 192.168.1.1/24 nghĩa là 24 bit đầu dành cho định danh mạng, còn lại 8 bit dành cho định danh host.

Việc đặt địa chỉ IP cho host phải tuân theo quy tắc:

- Các bit phần mạng không được phép đồng thời bằng 0. Ví dụ: 0.0.0.1/24 với phần định danh mạng là 0.0.0 và phần host là 1 là không hợp lệ.
- Địa chỉ có các bit phần định danh đồng thời bằng 0 hoặc đồng thời bằng 1 thì không được gán cho host. Hai địa chỉ IP này mặc dù hợp lệ nhưng được sử dụng cho các trường hợp riêng.

Số cổng có thể kết hợp với địa chỉ IP để tạo thành địa chỉ của tiến trình trên liên mạng. Loại địa chỉ này được viết theo cấu trúc **địa_chỉ_ip : số_cổng**. Loại địa chỉ này còn có tên gọi khác là địa chỉ điểm cuối (end-point).

Ví dụ, 192.168.1.1:80 là địa chỉ của một chương trình web server trên mạng cục bộ. Chương trình này đang chạy trên máy có IP 192.168.1.1 và chiếm cổng 80.

Chi tiết cách quản lý và sử dụng địa chỉ IP trong truyền thông mạng sẽ được trình bày trong bài học số 6.

c) Mặt nạ mạng con

Mặt nạ mạng con (subnet mask) là một dãy nhị phân dài 32 bit đi kèm với một địa chỉ IP để cho phép xác định được các bit dùng cho phần Network ID.

Điều này được thực hiện bằng phép toán AND địa chỉ IP với subnet-mask theo từng bit một.

Ví dụ: Xét địa chỉ IP 192.168.1.1 với subnet-mask là 255.255.255.0. Để xác định địa chỉ mạng của địa chỉ này, thực hiện AND 192.168.1.1 với 255.255.255.0.

Các subnet mask chuẩn của các địa chỉ lớp A, B, C:

- Lớp A: 255.0.0.0
- Lớp B: 255.255.0.0
- Lớp C: 255.255.255.0

Một cách khác để chỉ định số bit thuộc về phần định danh mạng trong địa chỉ IP là sử dụng số **prefix - length**. Số prefix - length được viết ngay sau địa chỉ IP và ngăn cách bởi dấu "/". Giá trị của prefix length cho biết có bao nhiêu bit đầu trong địa chỉ IP thuộc về phần định danh mạng. Số bit còn lại thuộc về phần định danh host. Ví dụ:

- 192.168.1.1/24 - 24 bit đầu tiên (3 octet đầu) dùng để định danh mạng, 8 bit (1 octet) còn lại cho định danh host
- 172.168.2.1/16 - 16 bit đầu tiên (2 octet đầu) dùng để định danh mạng, 16 bit (2 octet) tiếp theo cho định danh host
- 10.0.0.8/8 - 8 bit (1 octet) đầu cho định danh mạng, 24 bit (3 octet) tiếp theo cho định danh host.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) Chạy lại chương trình đã viết ở bài thực hành số 3.1, sau đó sử dụng phần mềm Wireshark bắt các gói tin UDP do chương trình gửi / nhận. Sử dụng kiến thức về cấu trúc header của UDP để tính ra giá trị các trường: cổng nguồn, cổng đích, và dữ liệu (SDU) của các gói tin này.
2) Chạy lại chương trình đã viết ở bài thực hành số 4.1, sau đó sử dụng phần mềm Wireshark bắt các gói tin TCP do chương trình gửi / nhận. Sử dụng kiến thức về cấu

trúc header của TCP để tính ra giá trị các trường: cổng nguồn, cổng đích, và dữ liệu của các gói tin này.

3) Sử dụng phần mềm Wireshark bắt các gói tin IP trong khi chạy phần mềm đã làm ở phần 3.1 và 4.1. Sử dụng kiến thức về cấu trúc header của IP để xác định địa chỉ IP nguồn, địa chỉ IP đích, tên giao thức đang khai thác dịch vụ IP.

Câu hỏi lý thuyết

1) Tại sao cần phải có giao thức khi hai thiết bị đã kết nối được với nhau về mặt vật lý?

Gợi ý: Nếu không có giao thức, dữ liệu có thể được truyền đi, nhưng điều gì khiến hai bên "hiểu đúng" nội dung gửi - nhận?

2) Nếu một ứng dụng yêu cầu tốc độ truyền cao nhưng không quan tâm đến mất mát dữ liệu, giao thức nào nên được sử dụng - TCP hay UDP? Tại sao?

Gợi ý: So sánh giữa độ tin cậy và chi phí truyền thông của hai giao thức.

3) Một server chỉ có thể xử lý một kết nối TCP tại một thời điểm. Theo bạn, điều gì sẽ xảy ra nếu nhiều client cùng kết nối tới server đó cùng lúc? Làm thế nào để phân biệt các kết nối?

Gợi ý: Suy nghĩ về cách kết hợp giữa địa chỉ IP và số cổng để xác định phiên truyền thông.

4) Giao thức IP hoạt động theo nguyên tắc "best-effort delivery". Theo bạn, điều này ảnh hưởng như thế nào đến ứng dụng như video call hoặc truyền file?

Gợi ý: Xem xét ứng dụng nào chấp nhận mất mát gói, ứng dụng nào thì không.

5) Giao thức UDP không có cơ chế xác nhận (ACK). Làm thế nào một ứng dụng sử dụng UDP vẫn có thể đảm bảo việc gửi dữ liệu thành công?

Gợi ý: Liệu ứng dụng có thể tự xây dựng giao thức riêng phía trên UDP?

6) Một client gửi chuỗi "HELLO123" nhiều lần liên tiếp qua TCP mà không ngắt kết nối. Theo bạn, server sẽ nhận được bao nhiêu message riêng biệt?

Gợi ý: Liệu TCP có phân tách thành từng gói riêng hay không?

7) Vì sao địa chỉ IP không đủ để định danh một ứng dụng cụ thể trong thiết bị? Thông tin nào cần được bổ sung?

Gợi ý: Một máy tính có thể chạy nhiều ứng dụng cùng lúc — điều gì phân biệt được?

8) Giả sử bạn được yêu cầu thiết kế một giao thức đơn giản để client và server trò chuyện văn bản. Theo bạn, những yếu tố nào bắt buộc phải được xác định trước để giao tiếp thành công?

Gợi ý: Hãy liệt kê các quy ước như định dạng dữ liệu, thứ tự trao đổi, mã hóa...

BÀI 5. THIẾT BỊ MẠNG

Mục tiêu của bài học

Bài học nhằm giúp sinh viên nắm bắt vai trò, đặc điểm và cơ chế hoạt động của các thiết bị mạng cơ bản, từ đó có khả năng thiết lập và kiểm thử mô hình mạng LAN đơn giản trong môi trường mô phỏng. Cụ thể, sau khi hoàn thành bài học, sinh viên có thể:

Về kiến thức:

- Mô tả chức năng và nguyên lý hoạt động của các thiết bị mạng như NIC, Hub, Switch và Router.
- Phân biệt đặc điểm của địa chỉ MAC và địa chỉ IP, vai trò của mỗi loại trong truyền dữ liệu.
- Trình bày được các mô hình liên kết mạng vật lý phổ biến và ưu nhược điểm của từng mô hình.

Về kỹ năng:

- Sử dụng phần mềm Packet Tracer để thiết kế, cấu hình và kiểm tra hoạt động mạng LAN cơ bản.
- Cấu hình địa chỉ IP tĩnh cho các thiết bị đầu cuối trong mạng.
- Thực hiện kiểm tra kết nối và phân tích mô hình hoạt động của thiết bị thông qua các công cụ đơn giản như ping và trình duyệt web giả lập.

5.1. Thiết bị mạng

5.1.1. Khái niệm và đặc điểm của thiết bị mạng

Như trong bài học đầu tiên chúng ta đã thấy rằng hai máy tính (thiết bị đầu cuối) có thể sử dụng cáp xoắn để kết nối trực tiếp tạo ra một mạng máy tính đơn giản. Khoảng cách truyền dữ liệu của cáp xoắn là khoảng 100m. Mạng cơ bản này chỉ có khả năng kết nối đúng 2 máy tính. Như vậy, mạng máy tính kiểu này rất hạn chế và khả năng ứng dụng không cao. Do vậy cần có giải pháp mở rộng mạng máy tính.

Thiết bị mạng là loại thiết bị trung gian giúp mở rộng mạng máy tính.

Việc "mở rộng" mạng máy tính có thể hiểu theo nhiều cách:

- Tăng khoảng cách truyền tín hiệu,
- Tăng số lượng thiết bị có thể kết nối,
- Tăng khả năng vận hành của mạng,
- Ghép nối các mạng nhỏ thành mạng lớn.

Với từng mục tiêu sẽ có loại thiết bị mạng phù hợp.

Để tạo ra một mạng cục bộ với nhiều máy tính có thể cùng trao đổi dữ liệu, người ta cần đến một loại thiết bị riêng giúp ghép nối nhiều máy tính lại với nhau. Có hai loại thiết bị đáp ứng yêu cầu này là hub và switch (cùng phiên bản đơn giản của chúng là repeater và bridge).

Khi có nhiều mạng cục bộ riêng rẽ, người ta cần đến một loại thiết bị giúp ghép nối các mạng cục bộ riêng thành một mạng cục bộ lớn hơn. Khi cần kết nối mạng cục bộ với mạng diện rộng, người ta cũng cần loại thiết bị riêng. Loại thiết bị cho phép ghép nối các mạng riêng biệt lại với nhau được gọi là router.

Thiết bị mạng rất đa dạng. Thiết bị mạng có thể là các máy tính chuyên dụng với hệ điều hành riêng. Cũng có một số loại thiết bị mạng chỉ đơn thuần là các mạch điện tử để chuyển tiếp và khuếch đại tín hiệu.

Thiết bị mạng không cho phép cài đặt phần mềm ứng dụng. Do đó, người dùng đầu cuối không trực tiếp tương tác với thiết bị mạng. Đây là đặc điểm phân biệt thiết bị mạng với thiết bị đầu cuối.

Thiết bị mạng chỉ trung chuyển tín hiệu và dữ liệu. Nó không phải là nơi xuất phát hay đích đến của dữ liệu người dùng. Đây cũng là đặc điểm phân biệt nó với thiết bị đầu cuối.

Trên các thiết bị mạng cũng có các chương trình giao thức hoạt động phục vụ tương tác giữa các thiết bị mạng với nhau hoặc giữa thiết bị mạng với thiết bị đầu cuối.

5.1.2. Card giao diện mạng

a) Đặc điểm của card mạng

Card mạng (Network Interface Card, NIC) là thiết bị cho phép kết nối thiết bị đầu cuối (và một số thiết bị mạng) với môi trường truyền.

NIC không phải là một thiết bị trung gian để mở rộng mạng như khái niệm về thiết bị mạng. Tuy nhiên NIC lại đóng vai trò thiết yếu trong hoạt động của mạng máy tính, là điểm bắt đầu và kết thúc luồng truyền thông tin trong mạng máy tính. Vì vậy, nội dung về NIC được đưa vào bài học về các thiết bị mạng.

Trên thiết bị đầu cuối, NIC có thể được tích hợp sẵn trên bo mạch chính, kết nối qua khe mở rộng PCI, hoặc kết nối qua cổng USB.

D-Link
AirPlus G

Hình 5.1. Hai loại card giao diện mạng gắn trong PC qua ke PCI

Trên mỗi NIC của thiết bị có các chương trình giao thức mạng để trao đổi dữ liệu. Đây là những chương trình chạy trên phần cứng của NIC. Card mạng làm việc với cả dữ liệu và tín hiệu.

Đơn vị dữ liệu mà card mạng xử lý được gọi là **frame**. Về bản chất, frame là một chuỗi byte có định dạng tùy thuộc vào công nghệ mạng. Frame là dạng dữ liệu số gần sát nhất với tín hiệu vật lý theo ý nghĩa: các bit của frame sẽ được chuyển đổi thẳng thành tín hiệu vật lý, và tín hiệu vật lý có thể chuyển đổi thành frame. Các đơn vị thông tin số hóa khác phải chuyển về frame trước khi đưa lên đường truyền.

Loại *tín hiệu* mà card mạng làm việc bao gồm tín hiệu điện, quang, và sóng vô tuyến. Phụ thuộc vào môi trường truyền, NIC có giao diện kết nối khác biệt nhau. Thông thường mỗi NIC chỉ có thể kết nối với một trong số các môi trường cáp xoắn, cáp quang, cáp đồng trục hoặc wifi.

Các chức năng chính của card mạng bao gồm:

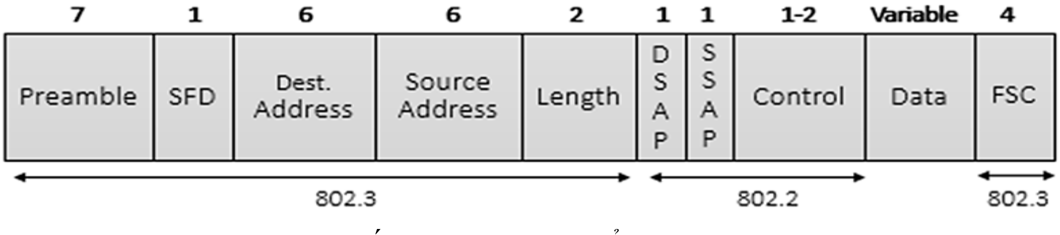
- Chuẩn bị dữ liệu đưa lên mạng: card mạng phải chuyển đổi chuỗi byte của frame thành tín hiệu.
- Đưa tín hiệu lên môi trường truyền;
- Thu tín hiệu từ môi trường truyền;
- Chuyển tín hiệu về dạng số: tín hiệu thu được cần chuyển về dạng chuỗi byte frame để có thể xử lý trên thiết bị đầu cuối hoặc thiết bị mạng.
- Kiểm soát luồng dữ liệu giữa máy tính và môi trường truyền.

Mỗi card mạng có một địa chỉ duy nhất dùng để phân biệt Card mạng này với Card mạng khác trên mạng, được gọi là *địa chỉ vật lý* hoặc *địa chỉ MAC.*

b) Cấu trúc chung của frame

Như ở trên đã nói, đơn vị dữ liệu mà card mạng xử lý có tên gọi là **frame**. Thông thường, mỗi frame bao gồm hai phần: phần đầu là một chuỗi byte có cấu trúc cố định, gọi là frame *header*; phần sau là một chuỗi byte do chương trình giao thức IP gửi xuống, gọi là *payload*.

Dưới đây là cấu trúc frame Ethernet chuẩn (IEEE 802.3).



Hình 5.2. Cấu trúc Frame chuẩn của IEEE 802.3

Thông tin chi tiết về các trường của frame Ethernet 802.3 dưới đây được đưa ra để tham khảo thêm. Sinh viên không cần ghi nhớ hoặc hiểu chi tiết về tất cả các trường. Những trường cần quan tâm là Destination Addresss và Source Address.

Ý nghĩa của các trường như sau:

- Preamble (Tiền amble): Đây là một chuỗi bit đặc biệt (7 byte trong đó mỗi byte có giá trị là 10101010 và byte cuối cùng là 10101011) được sử dụng để đồng bộ hóa khung dữ liệu và giúp thiết bị nhận xác định được đầu khung.
- Start of Frame Delimiter (SFD - Ngăn cách bắt đầu khung): Đây là một byte đặc biệt có giá trị là 10101011, được sử dụng để chỉ ra kết thúc của preamble và bắt đầu của khung dữ liệu.
- Destination Address (Địa chỉ MAC đích): Đây là trường 6 byte (48 bit) mô tả địa chỉ vật lý của thiết bị đích mà khung dữ liệu được gửi tới. Địa chỉ MAC (Media Access Control) của thiết bị nhận được sử dụng ở trường này.
- Source Address (Địa chỉ MAC nguồn): Đây là trường 6 byte (48 bit) mô tả địa chỉ vật lý của thiết bị gửi khung dữ liệu. Địa chỉ MAC của thiết bị gửi được sử dụng ở trường này.
- Type / Length: Đây là trường 2 byte (16 bit) được sử dụng để xác định kiểu dữ liệu của khung Ethernet. Nếu giá trị trong trường này nhỏ hơn hoặc bằng 1500 (0x05DC trong hex), nó sẽ đại diện cho trường Length, chỉ định độ dài của phần dữ liệu. Ngược lại, nếu giá trị lớn hơn 1500, nó sẽ chỉ định kiểu dữ liệu của Ethernet (ví dụ: IPv4, ARP, IPv6).
- Payload / Data (Dữ liệu): Đây là trường chứa dữ liệu thực tế được truyền qua mạng. Độ dài của trường này có thể thay đổi tùy thuộc vào giá trị trong trường EtherType/Length.
- Frame Check Sequence (FCS - Tính toán kiểm tra khung): Đây là trường 4 byte (32 bit) chứa giá trị kiểm tra khung dữ liệu được tính toán bằng một thuật toán như CRC (Cyclic Redundancy Check). Nó được sử dụng để kiểm tra tính toàn vẹn của dữ liệu trong khung.

c) Địa chỉ MAC

Địa chỉ MAC là một số nguyên 48 bit (6 byte) dùng để định danh cho card mạng và được sử dụng để truyền thông trong mạng LAN. Địa chỉ MAC được thể hiện bởi 12 chữ số Hexa và thường được viết tách thành cặp. Ví dụ: 20-32-33-CA-F4-45 (định dạng phổ biến trên Windows), 20:32:33:CA:F4:45 (định dạng phổ biến trên Linux/macOS).

Mỗi địa chỉ MAC được chia làm 2 phần:

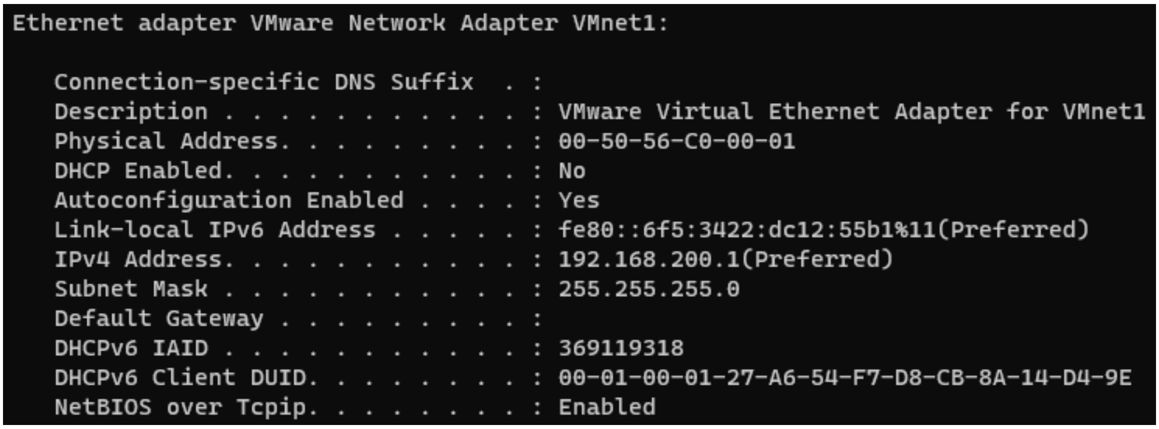
- Mã định danh của nhà sản xuất card mạng, do IEEE cấp chiếm 24 bit đầu tiên, thường được gọi tắt là OUI (Organizationally Unique Identifier);
- Số sê-ri duy nhất được nhà sản xuất gán cho từng card mạng cụ thể chiếm 24 bit còn lại, thường được gọi là NIC Specific / Device ID.

Một địa chỉ MAC đặc biệt, FF:FF:FF:FF:FF:FF, không được gán cho card mạng nào. Nó được dùng làm địa chỉ quảng bá trong mạng LAN. Chúng ta sẽ tìm hiểu vai trò của địa chỉ MAC đặc biệt này trong bài học về truyền thông tin trong mạng LAN.

Địa chỉ MAC được gán sẵn trong ROM của card mạng. Vì vậy, nó không thể thay đổi được giá trị và trở thành thông tin đặc trưng cho từng card mạng cụ thể.

Ngoài địa chỉ MAC, mỗi card mạng cũng được hệ điều hành gán cho một địa chỉ **địa chỉ IP**. Địa chỉ IP không gắn cứng với card mạng mà do hệ điều hành quản lý. Vì vậy, địa chỉ IP cũng được gọi là địa chỉ logic. Địa chỉ IP giúp thiết bị đầu cuối thực hiện truyền thông liên mạng.

Trên hệ điều hành windows có thể xem địa chỉ IP và địa chỉ MAC của một card mạng qua lệnh **ipconfig /all** chạy từ giao diện dòng lệnh.



```
Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VMware Virtual Ethernet Adapter for VMnet1
   Physical Address. . . . . . . . . : 00-50-56-C0-00-01
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::6f5:3422:dc12:55b1%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.200.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 369119318
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-27-A6-54-F7-D8-CB-8A-14-D4-9E
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Hình 5.3. Thông tin về card mạng khi chạy lệnh ipconfig

Chi tiết về các sử dụng địa chỉ MAC và địa chỉ IP trong quá trình truyền dữ liệu sẽ được trình bày ở bài học số 6 “Truyền thông tin trong mạng”.

5.1.3. Bộ tập trung tín hiệu

Khi tín hiệu truyền ở khoảng cách lớn có thể sẽ bị suy hao (ví dụ, do điện trở của đường truyền cáp đồng). Do đó để mở rộng mạng cần có thiết bị cho phép phục hồi hoặc khuếch đại tín hiệu.

Bộ tập trung tín hiệu (network hub) là loại thiết bị mạng LAN với nhiều cổng cho phép kết nối với nhiều thiết bị khác và làm nhiệm vụ *chuyển tiếp tín hiệu* giữa tất cả các thiết bị khác kết nối với nó.

Hub là một loại thiết bị đã lỗi thời. Hiện này hub không còn được sản xuất và sử dụng nữa. Chúng ta đề cập đến hub trong bài học này vì lý do sư phạm, cũng như để làm nổi bật vai trò của loại thiết bị thay thế cho hub - thiết bị switch - ở phần sau.

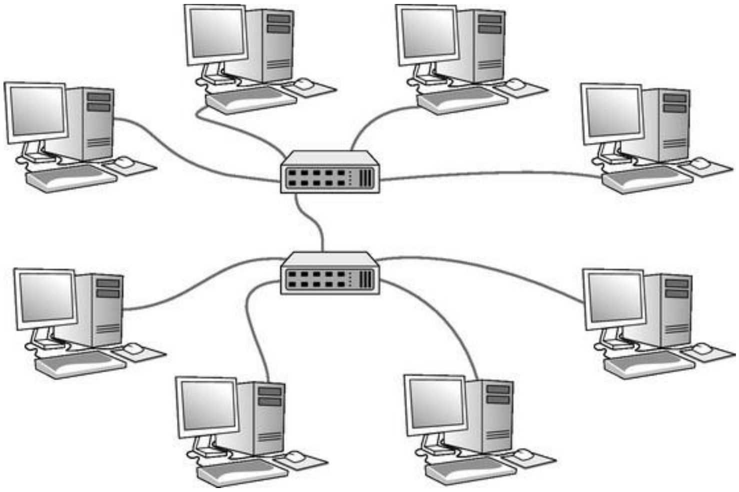
a) Kết nối hub trong mạng LAN

Mỗi hub bao gồm nhiều cổng RJ45. Mỗi cổng kết nối bằng cáp xoắn tới một thiết bị đầu cuối hoặc kết nối tới hub khác. Để kết nối thiết bị đầu cuối với hub cần dùng cáp thẳng. Để kết nối hub với nhau cần dùng cáp chéo.

NETGEAR
GS308
1
2
3
4
5
6
7
8

Hình 5.4. Một hub nhiều cổng của NetGear

Như vậy, hub cho phép mở rộng mạng LAN cả về phạm vi và số lượng thiết bị đầu cuối có thể tham gia.



Hình 5.5. Kết nối các thiết bị trong LAN với hub

b) Hoạt động của hub

Hoạt động của hub có một số đặc điểm cần lưu ý:

- Hub *chuyển tiếp tín hiệu* mà nó nhận được từ một cổng tới tất cả các cổng còn lại. Tùy thuộc vào loại hub, tín hiệu còn có thể được khuếch đại hoặc tái tạo. Hub không thực hiện biến đổi trên dữ liệu.
- Nếu một thiết bị đầu cuối gửi tín hiệu lên đường truyền tới hub, tất cả các thiết bị khác trong mạng sẽ nhận được tín hiệu. Phương thức truyền này được gọi là *truyền quảng bá (broadcast)*.
- Các thiết bị đầu cuối phải chuyển đổi tín hiệu thành frame rồi so sánh với địa chỉ MAC trong đó để xem frame dữ liệu có dành cho mình hay không. Nếu frame không dành cho mình, card mạng sẽ loại bỏ frame và không xử lý tiếp.

- Nếu có nhiều thiết bị cùng gửi tín hiệu lên đường truyền, tín hiệu có thể bị xung đột với nhau dẫn đến làm hỏng quá trình truyền thông. Do vậy, người ta phải đưa ra một giao thức để kiểm soát truy cập đường truyền chia sẻ giữa các thiết bị gọi là *CSMA/CD* (Carrier-Sense Multiple Access With Collision Detection).

c) Ưu nhược điểm

Về ưu điểm, hub có giá thành rẻ, dễ lắp đặt, không làm tăng thời gian xử lý của mạng, có thể mở rộng mạng cả về khoảng cách và số lượng thiết bị đầu cuối.

Về nhược điểm, hub không thể phân biệt giữa tín hiệu thực tế và nhiễu, không thể làm giảm lưu lượng mạng hoặc tắc nghẽn, và bị giới hạn về số lượng hub có thể được triển khai.

Trước đây hub được sử dụng rất phổ biến trong mạng LAN. Hiện nay hub không còn được sử dụng phổ biến nữa. Vai trò của nó trong LAN được thay thế bởi bộ chuyển mạch (switch).

5.1.4. Bộ chuyển mạch

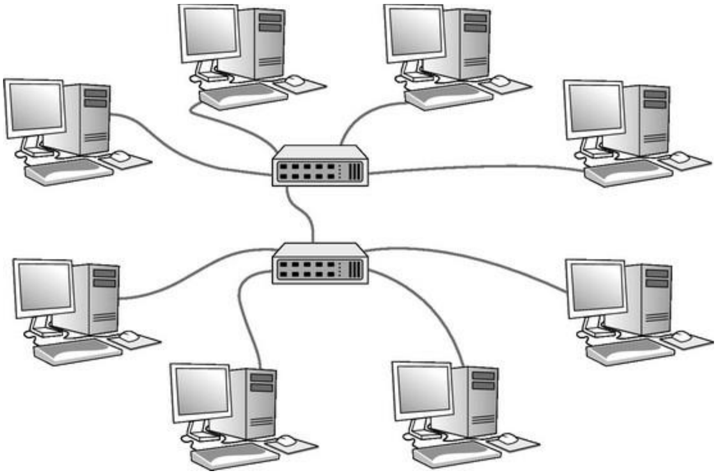
Bộ chuyển mạch (switch) là loại thiết bị mạng LAN với nhiều cổng cho phép kết nối với nhiều thiết bị khác và làm nhiệm vụ *chuyển tiếp frame dữ liệu* giữa các thiết bị khác kết nối với nó. Switch cho phép mở rộng mạng máy tính theo cách cho phép nhiều thiết bị kết nối để tạo thành một mạng LAN lớn.

a) Kết nối switch trong mạng LAN

Mỗi switch bao gồm nhiều cổng RJ45. Các switch giá rẻ dùng trong gia đình thường có 5 đến 8 cổng. Các switch chuyên dụng trong mạng doanh nghiệp thường có 12, 24 hoặc 48 cổng.

Hình 5.6. Switch nhiều cổng của HP

Mỗi cổng switch có thể kết nối bằng cáp xoắn tới một thiết bị khác. Switch có thể kết nối với thiết bị đầu cuối, với router, với hub, với switch khác. Để kết nối switch với thiết bị đầu cuối, router và hub cần dùng cáp thẳng. Để kết nối các switch với nhau cần dùng cáp chéo. Như vậy, các thiết bị đầu cuối và các switch có thể kết nối với nhau để tạo thành một mạng LAN lớn.



Hình 5.7. Kết nối máy tính với switch và kết nối switch với nhau

b) Hoạt động của switch

Switch chuyển tiếp frame dữ liệu mà nó nhận được từ một cổng tới một trong số các cổng còn lại căn cứ vào địa chỉ MAC của thiết bị gửi và nhận trong frame.

Các bước trong hoạt động của switch như sau:

(1) Khi nhận được tín hiệu đến từ một cổng, switch chuyển đổi tín hiệu thành frame dữ liệu. Quá trình chuyển đổi này tương tự như trên card mạng.

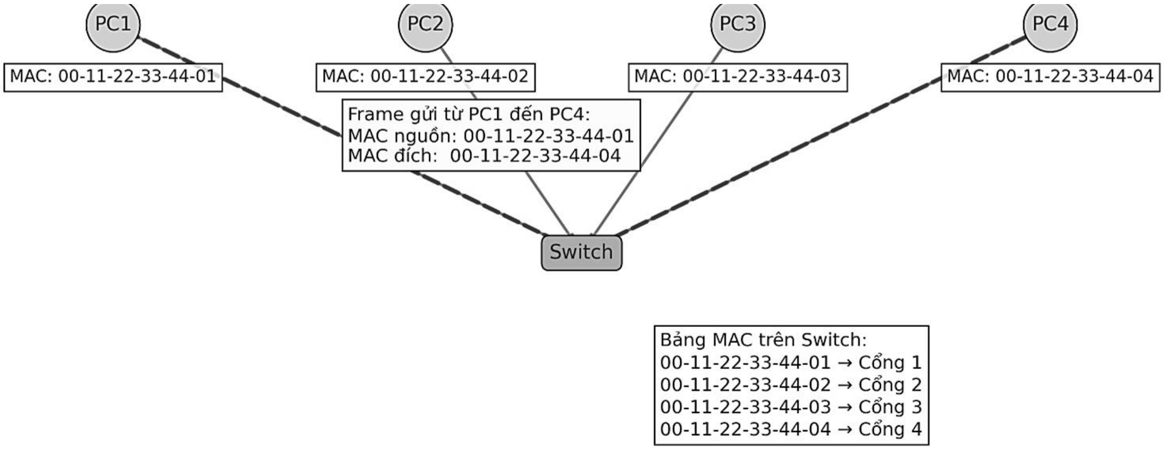
(2) Switch tách địa chỉ MAC nguồn và đích từ frame và tra cứu trong một bảng thông tin, gọi là *bảng chuyển mạch*, để xem thiết bị đích gắn với cổng nào. Bảng chuyển mạch chứa danh sách địa chỉ MAC của các thiết bị kết nối với switch và số cổng tương ứng của thiết bị đó.

Bảng chuyển mạch được lưu trong một bộ nhớ đặc biệt gọi là CAM (Content Addressable Memory), một loại bộ nhớ cho phép tìm kiếm rất nhanh theo nội dung. Nhờ CAM, switch có thể xử lý và chuyển tiếp frame hiệu quả gần như tức thời.

(3) Nếu địa chỉ MAC nguồn chưa có trong bảng chuyển mạch, switch sẽ đưa địa chỉ mới này vào. Quá trình này gọi là học địa chỉ MAC, giúp switch tự động xây dựng bảng chuyển mạch dựa trên dữ liệu đi qua.

(4) Khi xác định được cổng đích, switch chuyển tiếp frame sang cổng đó và chuyển lại thành tín hiệu để truyền đi. Nếu không thể xác định cổng đích, switch sẽ gửi broadcast frame đến tất cả các cổng (trừ cổng gửi), giống như hub.

Ví dụ, một mạng LAN nhỏ có 4 thiết bị đầu cuối (PC1 đến PC4) cùng kết nối tới switch. Mỗi thiết bị đầu cuối đều có địa chỉ MAC riêng biệt. Giả sử PC1 muốn gửi thông tin tới PC4. Frame gửi từ PC1 đến PC4 có: Địa chỉ MAC nguồn: 00-11-22-33-44-01, Địa chỉ MAC đích: 00-11-22-33-44-04.Switch tra bảng MAC để xác định PC4 được gắn với cổng số 4, và chỉ chuyển tiếp frame qua đúng cổng này, thay vì gửi tới tất cả thiết bị.


PC1
PC2
PC3
PC4
MAC: 00-11-22-33-44-01
MAC: 00-11-22-33-44-02
MAC: 00-11-22-33-44-03
MAC: 00-11-22-33-44-04
Frame gửi từ PC1 đến PC4:
MAC nguồn: 00-11-22-33-44-01
MAC đích: 00-11-22-33-44-04
Switch
Bảng MAC trên Switch:
00-11-22-33-44-01 → Cổng 1
00-11-22-33-44-02 → Cổng 2
00-11-22-33-44-03 → Cổng 3
00-11-22-33-44-04 → Cổng 4

Hình 5.8. Minh họa cơ chế chuyển mạch với địa chỉ MAC của switch

Như vậy, switch cho phép tách biệt giao thông trên từng phần của mạng, qua đó giúp tăng hiệu suất của mạng. Tín hiệu từ máy tính đi tới switch chỉ được chuyển tiếp tới thiết bị đích chứ không lan truyền trên toàn bộ mạng, do đó khả năng xung đột tín hiệu ít xảy ra hơn ngay cả khi có nhiều thiết bị đầu cuối cùng kết nối với switch.

Trong quá trình chuyển tiếp frame, switch có thể sử dụng một trong hai nguyên tắc:

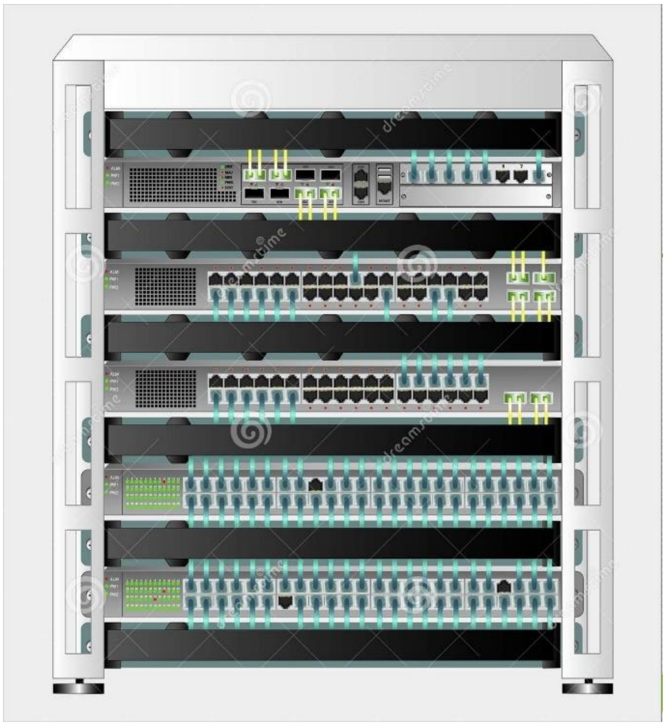
- **Lưu và chuyển tiếp** (Store and Forward): switch chờ nhận đủ frame thì mới chuyển tiếp; nếu chưa nhận đủ thì lưu lại cho đến khi đủ; nếu frame lỗi thì không phát. Cách này an toàn hơn với dữ liệu nhưng có độ trễ lớn.
- **Cut-Through**: switch bắt đầu chuyển tiếp frame ngay sau khi nhận ra địa chỉ MAC mà không cần chờ cho đủ frame dữ liệu. Cách này có độ trễ nhỏ nhưng có thể để lọt các frame lỗi.

c) Ứng dụng

Switch là thiết bị mạng LAN phổ biến nhất hiện nay. Nó được sử dụng trong mọi mạng LAN, từ gia đình đến doanh nghiệp.

Trong gia đình thường dùng các loại switch giá rẻ với 3 đến 8 cổng. Loại switch này thường được gọi là Unmanaged switch, chỉ cần cắm là chạy và không cần (và cũng không cho phép) các cấu hình nâng cao. Các loại switch này không chịu được lưu lượng mạng lớn và không cần bảo dưỡng đặc biệt.

Trong doanh nghiệp sử dụng những loại switch chuyên dùng với 12, 24 hoặc 48 cổng với giá thành rất cao. Loại switch này thuộc nhóm Managed switch. Loại switch này cung cấp khả năng cấu hình các dịch vụ nâng cao đặc biệt cho LAN như VLAN (chia nhỏ mạng LAN thành các mạng logic), QoS (ưu tiên lưu lượng theo dịch vụ), STP (ngăn ngừa vòng lặp mạng khi dùng nhiều switch). Loại switch chuyên dụng này thường được gắn vào tủ thiết bị (gọi là tủ rack) để bảo vệ và cố định các switch.



Hình 5.9. Một tủ rack với nhiều switch gắn bên trong

5.1.5. Thiết bị router

Bộ định tuyến (router) là loại thiết bị có nhiều cổng dùng để *ghép nối các mạng* với nhau thành một *liên mạng* (internet, lưu ý chữ i thường).

Khác biệt với các thiết bị đã xem xét ở trên, router không có nhiệm vụ ghép nối các thiết bị trong mạng LAN. Thay vào đó, router có nhiệm vụ ghép nối các mạng với nhau. Router có thể ghép nối các mạng LAN với nhau hoặc ghép nối mạng LAN với mạng WAN. Qua đó router giúp tạo ra một liên mạng lớn.

Hình 5.10. Hình ảnh một router của Cisco

a) Cấu tạo

Trong số các thiết bị mạng, router là loại thiết bị phức tạp nhất.

Về cấu tạo, *router là một máy tính chuyên dụng*. Nó có đầy đủ các thành phần gần tương tự như một máy tính thông thường (như ROM, RAM, bộ vi xử lý, bo mạch, các cổng kết nối). Tuy nhiên, nó không có các thành phần vào ra dữ liệu như máy tính thông thường (như chuột, bàn phím, màn hình). Do đó, người quản trị mạng phải dùng một máy tính khác để truy cập router khi cấu hình.

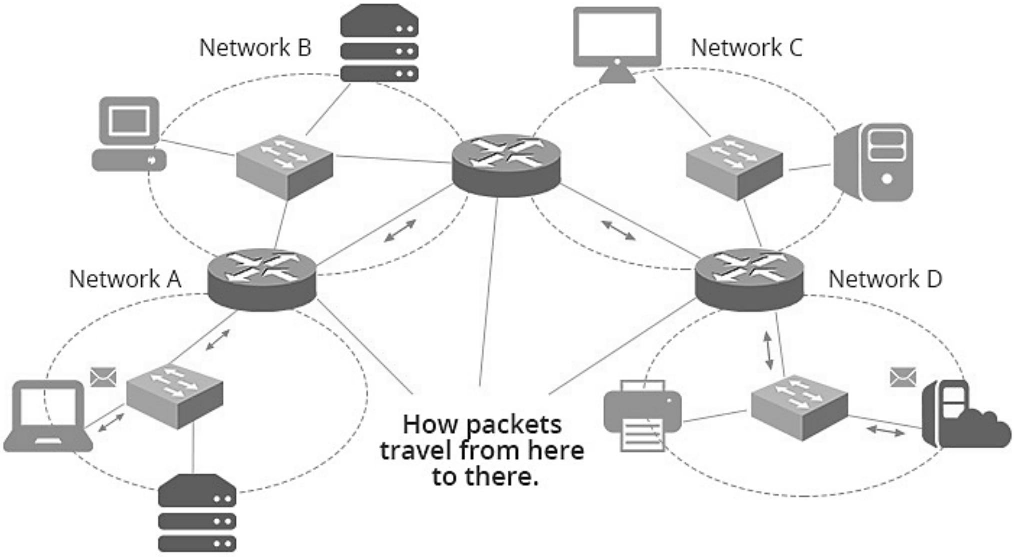
Trên router có sẵn nhiều thành phần để làm việc với mạng, gọi là các *module*. Router có các module để làm việc với các công nghệ LAN và WAN khác nhau. Ví dụ, router có thể chứa module để làm việc với mạng Ethernet, module làm việc với mạng cáp quang. Các module khác có thể được lắp thêm vào router hoặc gỡ bỏ.

Trên router cài đặt một *hệ điều hành riêng*. Hệ điều hành này cho phép cài đặt sẵn nhiều phần mềm giao thức khác nhau. Tuy có hệ điều hành riêng, router không cho phép cài đặt phần mềm ứng dụng.

b) Kết nối router trong mạng

Mỗi mạng LAN cơ bản thường được tạo ra từ một thiết bị switch kết nối với các thiết bị đầu cuối. Để các thiết bị đầu cuối trong LAN có thể trao đổi dữ liệu, mỗi thiết bị phải được gán một địa chỉ IP riêng biệt và *thuộc cùng một dải*.

Nếu có từ 2 mạng LAN như vậy và sử dụng dải địa chỉ IP khác nhau, chúng phải được kết nối thông qua thiết bị router thì thiết bị đầu cuối ở mạng này mới có thể trao đổi dữ liệu với thiết bị đầu cuối ở mạng khác.



Hình 5.11. Kết nối các mạng LAN bằng router

Một thiết bị router thường có nhiều cổng, mỗi cổng kết nối với một mạng khác nhau. Có thể hình dung mỗi cổng của router tương tự như một card mạng trên thiết bị đầu cuối. Mỗi cổng này có địa chỉ MAC và địa chỉ IP riêng phù hợp với từng mạng mà nó tham gia.

c) Hoạt động của router

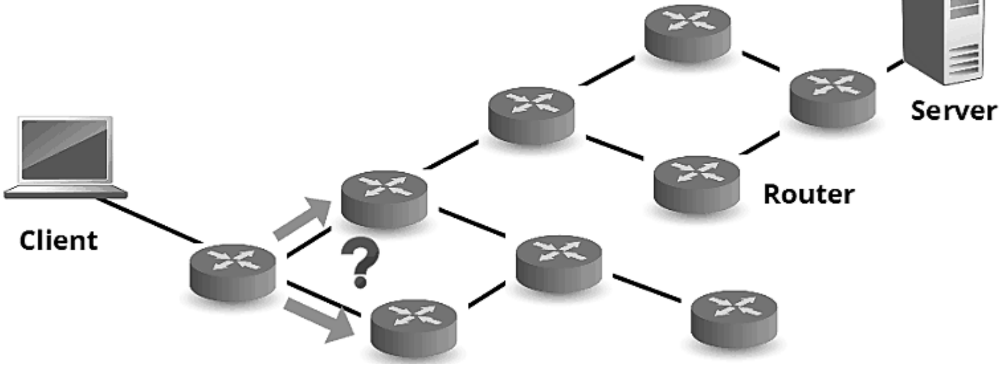
Để thực hiện được việc ghép nối các mạng khác biệt về công nghệ, router cần thực hiện hàng loạt công việc:

- Thu nhận tín hiệu từ mạng nguồn và chuyển đổi thành frame dữ liệu: vai trò này rất giống như của card mạng. Thực tế mỗi cổng của router có thể xem như một card mạng với địa chỉ MAC riêng.
- Tách gói tin IP khỏi frame dữ liệu và phân tích địa chỉ IP: frame dữ liệu chứa gói tin IP trong phần payload. Router cần địa chỉ IP của mạng đích để quyết định chuyển tiếp gói tin IP đó sang cổng nào.
- Chuyển tiếp gói tin IP sang mạng đích: quá trình này được thực hiện nhờ tìm kiếm trong một bảng thông tin về địa chỉ IP của các mạng đang kết nối với router.
- Đóng gói gói tin IP vào frame và đưa lên mạng đích tương ứng: nhiệm vụ này cũng giống như ở card mạng.

Trong số các nhiệm vụ trên, chuyển tiếp gói tin IP là quan trọng nhất và phức tạp nhất. Nó là mục đích của thiết bị router. Để xác định đích cho việc chuyển tiếp gói tin IP, router phải thực hiện một công việc gọi là *định tuyến*.

d) Định tuyến với router

Định tuyến (routing) là quá trình tìm kiếm và xác định đường đi tốt nhất trên một mạng máy tính để gói tin IP tới được đích. Sở dĩ phải có định tuyến là vì trong một liên mạng lớn, hai thiết bị đầu cuối thường sẽ nằm trong các mạng khác nhau. Giữa hai mạng này tồn tại rất nhiều thiết bị router (với nhiệm vụ ghép nối các mạng thành liên mạng). Khi đó, dữ liệu từ thiết bị đầu cuối này có thể có rất nhiều đường đi khác nhau tới thiết bị đầu cuối khác.



Hình 5.12. Có bao nhiêu đường đi cho dữ liệu từ client tới server?

Để làm được điều đó, thiết bị định tuyến cần phải dựa vào một bảng thông tin về tình trạng của mạng, gọi **bảng định tuyến** (routing Table). Bảng định tuyến được xây dựng một cách thủ công hoặc dựa trên các *giao thức định tuyến* (routing protocol). Nếu bảng định tuyến được xây dựng theo cách thủ công, quá trình định tuyến được gọi là *định tuyến tĩnh* (static routing). Định tuyến tĩnh đơn giản nhưng chỉ phù hợp với các liên mạng nhỏ với cấu trúc cố định. Nếu bảng định tuyến được xây dựng tự động bởi các giao thức định tuyến, chúng ta sẽ có *định tuyến động* (dynamic routing). Định tuyến động là bắt buộc đối với các mạng rất lớn (khi có rất nhiều đường đi tới mạng đích) hoặc khi mạng có cấu trúc biến đổi.

Giao thức định tuyến là các giao thức hoạt động trên router giúp các router chia sẻ thông tin về trạng thái của các mạng kết nối với chúng, qua đó giúp router "vẽ" ra tình trạng thực tế của mạng và tạo ra bảng định tuyến. Có rất nhiều giao thức định tuyến khác nhau được cài đặt sẵn trong router như RIP, IGRP, EIGRP, OSPF, IS-IS, v.v..

e) Ứng dụng

Router là thiết bị mạng phổ biến hàng đầu, cả trong LAN và trong WAN. Có thể nói rằng, nếu không có router thì không thể tạo ra liên mạng và không thể có mạng Internet.

Trong mạng gia đình thường sử dụng loại router nhỏ giá rẻ tích hợp cùng nhiều chức năng khác (như wifi access point, ethernet switch) trong cùng một thiết bị. Loại router này kết nối mạng LAN của gia đình với mạng WAN của nhà cung cấp dịch vụ Internet (Internet Service Provider, ISP).

Trong mạng doanh nghiệp, router là thiết bị quan trọng số 1 để ghép nối rất nhiều mạng LAN nhỏ của các bộ phận thành một mạng LAN lớn, qua đó tăng hiệu suất hoạt động và tính an toàn của mạng. Router cũng dùng để kết nối hệ thống LAN với đường truyền WAN cung cấp Internet băng thông cao cho doanh nghiệp.

Router cũng dùng để kết nối rất nhiều mạng WAN khác nhau của ISP để tạo ra hệ thống liên mạng trải rộng trong một vùng lãnh thổ, cũng như tạo ra mạng toàn cầu Internet.

5.1.6. Hạ tầng mạng gia đình

Hệ thống mạng trong gia đình có thể xem là một mạng cục bộ LAN tương đối đơn giản kết nối các thiết bị và cung cấp kết nối Internet. Dưới đây là mô tả chi tiết về cấu trúc một hệ thống mạng LAN trong một gia đình thông thường.

1. Thiết bị truy cập Internet (Internet Access Device):

- Thường là modem cáp quang, DSL, hoặc router.

- Kết nối với dịch vụ Internet của nhà cung cấp qua cáp quang hoặc DSL.

- Có thể tích hợp sẵn chức năng DHCP để cung cấp cấu hình IP động cho các thiết bị trong mạng gia đình.

2. Router:

- Router làm nhiệm vụ chuyển tiếp dữ liệu giữa mạng LAN trong gia đình và Internet.

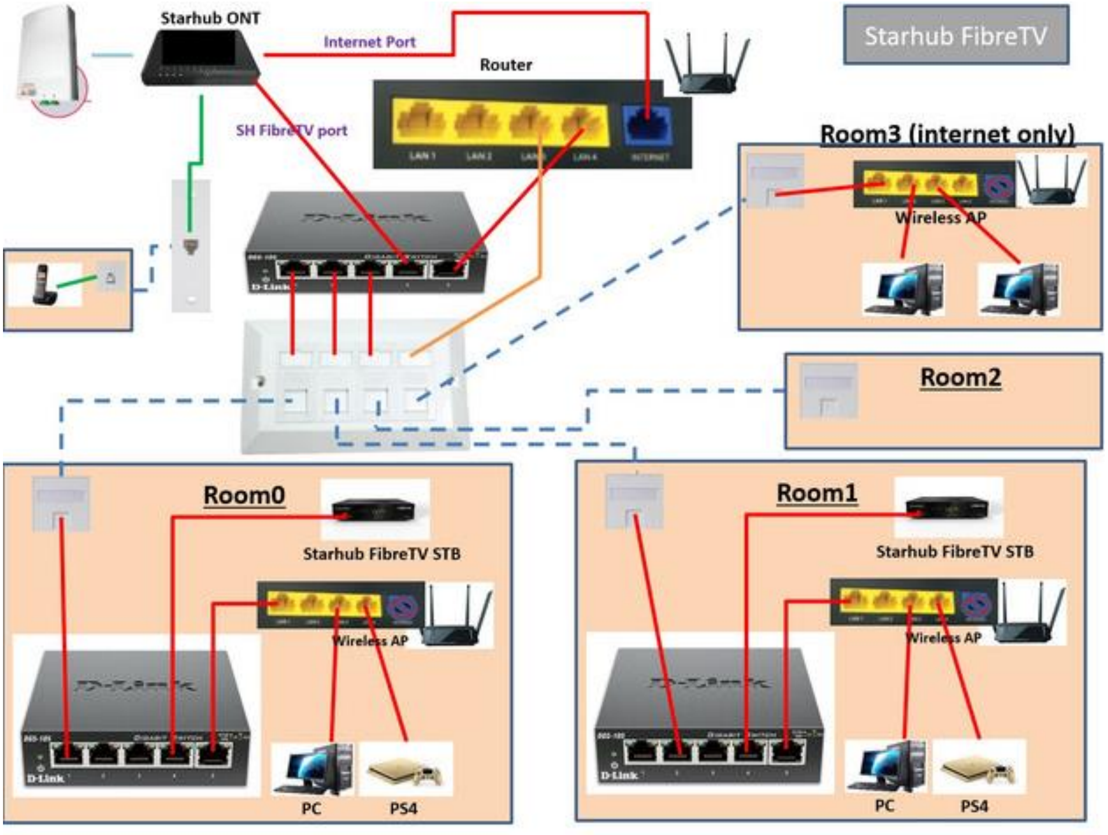
- Có thể có tích hợp sẵn trong thiết bị truy cập Internet hoặc sử dụng một router riêng biệt.

- Router cũng có khả năng chia sẻ địa chỉ IP public thành nhiều địa chỉ IP private sử dụng Network Address Translation (NAT).

3. Switch:

- Switch được sử dụng để kết nối các thiết bị mạng trong gia đình với nhau, chẳng hạn như máy tính, máy in, camera IP, TV thông minh, và thiết bị IoT (Internet of Things).

- Switch LAN giúp tạo một mạng nội bộ nhanh chóng và ổn định bằng cách chuyển tiếp gói dữ liệu dựa trên địa chỉ MAC của các thiết bị.


Starhub ONT
Internet Port
Router
SH FibreTV port
Starhub FibreTV
Room3 (internet only)
Wireless AP
Room2
Room0
Starhub FibreTV STB
Wireless AP
PC
PS4
Room1
Starhub FibreTV STB
Wireless AP
PC
PS4

Hình 5.13. Minh họa hạ tầng mạng gia đình

4. Máy tính cá nhân:

- Các máy tính trong gia đình được kết nối vào switch qua cáp Ethernet hoặc kết nối không dây Wi-Fi (sử dụng một access point hoặc tích hợp trong router).

- Máy tính cá nhân có thể chạy các hệ điều hành khác nhau và được sử dụng cho nhiều mục đích, bao gồm lướt web, gửi email, xem video, và chơi game trực tuyến.

5. Thiết bị di động (Mobile Devices):

- Các thiết bị di động như điện thoại thông minh, máy tính bảng và laptop có khả năng kết nối vào mạng Wi-Fi gia đình.

- Chúng sử dụng giao thức Wi-Fi để kết nối với router hoặc access point để truy cập Internet và tài nguyên mạng.

6. Máy in mạng (Network Printer):

- Máy in mạng được kết nối vào switch hoặc router qua cổng Ethernet hoặc Wi-Fi.

- Cho phép các thiết bị trong gia đình in ấn từ xa thông qua mạng LAN.

7. Thiết bị thông minh (Smart Devices):

- Các thiết bị thông minh như đèn chiếu sáng, ổ cắm thông minh, camera IP, và thiết bị điều khiển tùy chỉnh có thể được kết nối vào mạng LAN để điều khiển và quản lý từ xa thông qua ứng dụng điện thoại hoặc máy tính.

8. Firewall và Security Measures:

- Gia đình có thể sử dụng tường lửa (firewall) tích hợp trong router hoặc cài đặt tường lửa bảo mật để bảo vệ mạng LAN khỏi các mối đe dọa từ Internet.

- Các biện pháp bảo mật bao gồm cài đặt mật khẩu mạng Wi-Fi, cập nhật phần mềm định kỳ, và sử dụng mã hóa để bảo vệ dữ liệu mạng.

5.2. Cấu trúc liên kết mạng

5.2.1. Khái niệm cấu trúc liên kết mạng

Trong một mạng máy tính, việc kết nối các thiết bị không chỉ đơn thuần là cắm dây hoặc thiết lập kết nối không dây. Thay vào đó, các thiết bị cần được tổ chức và sắp xếp theo một cấu trúc nhất định, giúp đảm bảo việc truyền tải dữ liệu hiệu quả, dễ quản lý, dễ mở rộng, và duy trì hoạt động ổn định của hệ thống mạng.

Cấu trúc liên kết mạng (Topology) là cách thức sắp xếp và kết nối các thiết bị mạng (như máy tính, máy in, router, switch...) thông qua các liên kết vật lý hoặc logic. Có thể hiểu đơn giản: Topology là "sơ đồ đường đi" của thông tin trong mạng.

Mỗi cấu trúc liên kết mạng sẽ xác định:

- Thiết bị nào kết nối với thiết bị nào?
- Dữ liệu sẽ đi theo đường nào để đến đúng đích?
- Cách xử lý khi có sự cố hoặc cần mở rộng hệ thống?

Cấu trúc liên kết có thể chia thành hai loại chính:

- Liên kết vật lý (Physical Topology): Mô tả cách thiết bị được nối với nhau bằng cáp, anten, sóng... Nhìn thấy được bằng mắt thường.

- Liên kết logic (Logical Topology): Mô tả cách dữ liệu di chuyển giữa các thiết bị trong mạng, bất kể sơ đồ dây dẫn thực tế.

Việc lựa chọn mô hình kết nối thiết bị ảnh hưởng trực tiếp đến:

- Hiệu suất mạng: Dữ liệu đi đường xa hay gần, dễ bị tắc nghẽn hay không.
- Khả năng mở rộng: Thêm thiết bị mới dễ hay phức tạp.
- Độ tin cậy: Khi một thiết bị hỏng thì toàn mạng có ngừng hoạt động không?
- Chi phí đầu tư: Mỗi mô hình đòi hỏi chi phí cáp, thiết bị trung tâm, bảo trì khác nhau.
- Ứng dụng thực tế: Ví dụ, mạng gia đình, mạng văn phòng nhỏ và mạng trung tâm dữ liệu thường chọn mô hình khác nhau.

Trong phần tiếp theo, chúng ta sẽ tìm hiểu các mô hình vật lý phổ biến nhất, bao gồm:

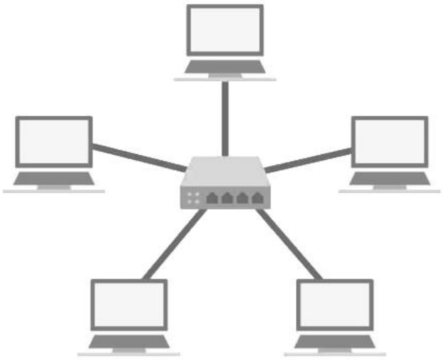
- Mô hình tuyến tính (Bus): Tất cả các thiết bị chia sẻ một đường truyền duy nhất.
- Mô hình vòng (Ring): Các thiết bị kết nối thành một vòng tròn khép kín.
- Mô hình hình sao (Star): Các thiết bị kết nối tập trung qua một thiết bị trung tâm như switch.
- Mô hình hình lưới (Mesh): Các thiết bị kết nối trực tiếp từng đôi một với nhau.
- Mô hình kết hợp (Hybrid): Kết hợp nhiều mô hình trên.

5.2.2. Cấu trúc hình sao

Khái niệm

Mô hình Mạng hình sao là mô hình mà ở đó các thiết bị đầu cuối được lần lượt kết nối với một thiết bị trung tâm. Thiết bị trung tâm này có thể là Hub hoặc Switch.

Hoạt động: Khi một máy có nhu cầu truyền dữ liệu, gói tin sẽ được truyền cho thiết bị trung tâm. Mỗi gói tin đều có định danh của thiết bị đầu cuối nhận tin. Thiết bị trung tâm sẽ chịu trách nhiệm chuyển tiếp gói tin đến thiết bị đầu cuối nhận tin. Ở đó, gói tin sẽ được xử lý.



Hình 5.14. Mô hình mạng hình sao

Ưu điểm

- Có độ tin cậy cao, nếu một dây mạng hoặc một thiết bị đầu cuối bị lỗi, không ảnh hưởng gì tới các thiết bị đầu cuối còn lại.

- Có hiệu năng hoạt động cao, không xảy ra xung đột nếu thiết bị trung tầm là Switch.
- Giá thành lắp đặt không cao.
- Dễ dàng lắp đặt.
- Khi thêm thiết bị đầu cuối hoặc mở rộng mạng sẽ không ảnh hưởng tới hoạt động của các thiết bị đầu cuối và mạng hiện tại.
- Dễ dàng xử lý sự cố.

Nhược điểm

- Sử dụng nhiều cáp hơn mạng tuyến tính và mạng hình tròn sẽ xem xét tiếp theo đây
- Phụ thuộc vào thiết bị mạng trung tâm. Nếu thiết bị mạng trung tâm bị hỏng, toàn bộ hệ thống mạng sẽ không sử dụng được
- Chi phí lắp đặt cao hơn mạng tuyến tính và mạng hình tròn, trong đó có chi phí không nhỏ cho thiết bị trung tâm
- Cần chi phí vận hành cho thiết bị trung tâm

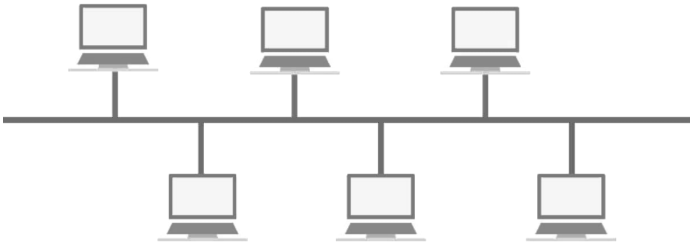
5.2.3. Cấu trúc tuyến tính

Khái niệm

Mô hình mạng tuyến tính là một kiểu mô hình mạng mà ở đó các thiết bị đầu cuối kết nối và sử dụng chung một đường cáp truyền tín hiệu. Đường cáp đó có thể là cáp xoắn hoặc cáp đồng trục. Hai đầu của sợi cáp được gắn một thiết bị đặc biệt có tên tiếng anh là Terminator. Terminator có chức năng ngăn việc mất hoặc suy giảm tín hiệu tại hai đầu của sợi cáp.

Hoạt động

Khi một thiết bị đầu cuối trong mạng tuyến tính muốn truyền giữ liệu, thiết bị đầu cuối đó sẽ gửi gói tin lên đường truyền tín hiệu. Gói tin đó có đính kèm địa chỉ định danh của thiết bị đầu cuối nhận thông tin. Tất cả các thiết bị đầu cuối trong mạng tuyến tính đều sẽ nhận được gói tin, tuy nhiên chỉ có thiết bị đầu cuối nào có trùng địa chỉ định danh ghi trên gói tin mới xử lý gói tin đó.



Hình 5.15. Mô hình mạng tuyến tính

Ưu điểm

- Là mô hình mạng dễ lắp đặt nhất.
- Hoạt động hiệu quả với số lượng ít thiết bị đầu cuối
- Số lượng cáp ít so với loại mô hình mạng khác
- Dễ dàng kết nối hoặc ngắt kết nối một thiết bị đầu cuối mà không ảnh hưởng tới các thiết bị khác
- Giá thành lắp đặt thấp so với các mô hình mạng khác
- Dễ dàng mở rộng bằng cách kết nối cáp của hai mạng tuyến tính với nhau

Nhược điểm

- Không hoạt động tốt với mạng nhiều thiết bị đầu cuối
- Tốc độ truyền dữ liệu chậm so khi so với các mô hình mạng khác
- Khó khăn trong giải quyết sự cố
- Tỉ lệ mất gói tin lớn
- Nếu sợi cáp tuyến tính bị hỏng hoặc đứt, mô hình mạng sẽ bị chia nhỏ
- Cứ thêm một thiết bị đầu cuối vào mạng tuyến tính, hiệu năng hoạt động của mạng sẽ bị giảm đi.

5.2.4. Cấu trúc vòng

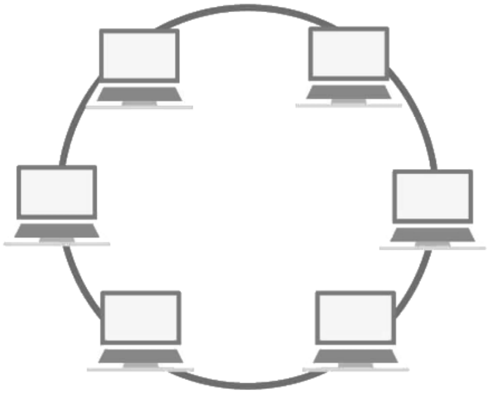
Khái niệm

Mô hình Mạng hình tròn là một kiểu mô hình mà ở đó các thiết bị đầu cuối kết nối với nhau thành một mạch tròn khép kín. Mỗi một thiết bị đầu cuối sẽ được kết nối chính xác với số lượng hai thiết bị đầu cuối liền kề. Nếu các thiết bị đầu cuối trong mạng hình tròn có số lượng lớn, các bộ lặp tín hiệu (repeater) sẽ được lắp thêm vào để đảm bảo tín hiệu trên vòng không bị suy yếu.

> Cấu trúc vòng là loại cấu trúc đặc thù của công nghệ mạng Token ring. Hiện nay loại công nghệ này không còn phổ biến. Các thiết bị dành cho công nghệ này đã không còn được sản xuất nữa.

Hoạt động

Khi một thiết bị đầu cuối muốn truyền dữ liệu, thiết bị cần đợi cho đến khi nhận được thẻ bài cho phép truyền tin. Thiết bị đầu cuối sẽ gửi gói tin lên đường truyền. Mỗi một gói tin đều có địa chỉ định danh của thiết bị đầu cuối nhận tin. Gói tin sẽ được các thiết bị đầu cuối khác chuyển tiếp trong mạng theo một chiều nhất định (đã được quy định trước) cho đến khi gói tin đến được thiết bị đầu cuối có trùng địa chỉ định danh trên gói tin. Khi ấy gói tin sẽ được thiết bị đầu cuối xử lý.



Hình 5.16. Mô hình mạng hình tròn

Ưu điểm

- Giảm xung đột khi truyền tin do các gói tin được chuyển đi theo một chiều nhất định.
- Khi một thiết bị đầu cuối được thêm vào mạng, hầu như sẽ không ảnh hưởng gì đến hiệu năng hoạt động của hệ thống mạng.
- Có giá thành lắp đặt, mở rộng vừa phải.
- Tốc độ truyền dữ liệu tốt.
- Tất cả các thiết bị đầu cuối đều có quyền ngang hàng trong việc truyền dữ liệu (dựa vào việc chuyển thẻ qua các thiết bị đầu cuối).

Nhược điểm

- Thiết bị đầu cuối cần nhận được thẻ bài để truyền tin.
- Nếu một thiết bị đầu cuối bị hỏng/bị ngắt nguồn/bị tê liệt sẽ làm toàn bộ hệ thống mạng bị tê liệt.
- Tốc độ truyền dữ liệu chậm hơn mạng tuyến tính.
- Khó khăn khi thêm bớt một thiết bị đầu cuối vào mạng, việc thêm thiết bị này có thể ảnh hưởng tới sự hoạt động của toàn mạng.
- Khó giải quyết sự cố.
- Khó mở rộng.

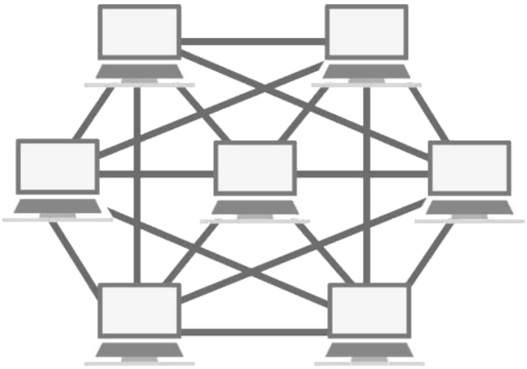
5.2.5. Cấu trúc lưới

Khái niệm

Mô hình Mạng hình lưới là một loại mô hình mà ở đó các thiết bị đầu cuối kết nối với nhau từng đôi một thông qua môi trường truyền dẫn. Các thiết bị đầu cuối không những đóng vai trò là thiết bị gửi nhận thông tin mà còn đóng vai trò là thiết bị tiếp sức cho gói tin của thiết bị đầu cuối khác.

Hoạt động

Khi một thiết bị đầu cuối có nhu cầu truyền dữ liệu, nó sẽ truyền trực tiếp dữ liệu trực tiếp dưới dạng gói tin cho thiết bị đầu cuối nhận thông qua môi trường truyền dẫn giữa hai thiết bị.



Hình 5.17. Mô hình mạng hình lưới

Ưu điểm

- Một thiết bị đầu cuối hỏng/lỗi không ảnh hưởng tới hệ thống mạng.
- Không có vấn đề về việc tranh chấp băng thông do các thiết bị đầu cuối được kết nối với nhau theo cặp.
- Rất dễ để xử lý sự cố.
- Khó mất gói tin.
- Tính bảo mật và tính riêng tư cao.
- Thêm bớt thiết bị đầu cuối không ảnh hưởng tới hệ thống mạng.
- Hoạt động hiệu quả với mọi tình thế phát sinh trong quá trình sử dụng.

Nhược điểm

- Giá thành lắp đặt cao nhất so với các loại mô hình mạng trên.
- Cài đặt phức tạp.
- Tiêu hao nhiều năng lượng hơn do các thiết bị đầu cuối đều đảm trách tiếp sức cho các gói tin được truyền đi trong mạng.
- Phức tạp trong vận hành, giá thành bảo trì cao.

Kết hợp của nhiều hơn một mô hình mạng trên tạo ra **mô hình mạng kết hợp.**

5.2.6. Cấu trúc liên kết và công nghệ mạng

Cấu trúc liên kết mạng (topology) phụ thuộc một phần vào công nghệ mạng đang được sử dụng.

Một số công nghệ mạng chỉ hỗ trợ hoặc hoạt động tối ưu với một số kiểu liên kết vật lý nhất định. Ví dụ: Công nghệ Ethernet dùng cáp xoắn đôi hoặc cáp đồng trục thời kỳ đầu (Ethernet 10Base2, 10Base5) dùng mô hình tuyến tính (bus). Tuy nhiên công nghệ Ethernet hiện đại (Fast Ethernet, Gigabit Ethernet...) dùng hình sao (star) với switch là thiết bị trung tâm. Trong công nghệ Token Ring, dữ liệu truyền theo mô hình logic hình vòng, dù vật lý có thể giống hình sao. Trong công nghệ Wi-Fi hầu hết sử dụng mô hình hình sao mở rộng (extended star) với access point là trung tâm nhưng về mặt logic lại hoạt động theo kiểu quảng bá (broadcast), gần giống bus.

Ở mức độ logic (truyền dữ liệu), cấu trúc liên kết mạng không phụ thuộc vào công nghệ sử dụng. Nhiều công nghệ có thể hoạt động trên nhiều kiểu topology vật lý, vì cấu trúc logic phụ thuộc vào cách truyền dữ liệu và thiết kế phần mềm mạng, ví dụ: Ethernet hình sao vật lý nhưng hoạt động logic gần giống bus nếu dùng hub (tín hiệu được broadcast); Wi-Fi có thể hỗ trợ nhiều điểm truy cập (multi-AP), tạo thành liên kết mesh logic nếu được cấu hình phù hợp.

Như vậy, công nghệ mạng ảnh hưởng đến khả năng triển khai topology, đặc biệt là về mặt vật lý. Tuy nhiên, trong thiết kế mạng hiện đại, người quản trị có thể chọn lựa topology phù hợp bằng cách kết hợp công nghệ, thiết bị, cấu hình phần mềm.

Do đó, có thể nói, cấu trúc liên kết vật lý bị ràng buộc bởi công nghệ phần cứng, còn cấu trúc liên kết logic thì linh hoạt hơn và do người thiết kế quyết định.

5.3. Thực hành - mô phỏng mạng với Packet Tracer

Mục tiêu:

- Hiểu và vận dụng kiến thức về thiết bị mạng: NIC, Switch.
- Thiết lập mô hình mạng LAN đơn giản sử dụng Switch và các máy tính đầu cuối.
- Cấu hình địa chỉ IP tĩnh và kiểm tra kết nối trong mạng LAN.

Mô tả bài thực hành:

Sinh viên sử dụng phần mềm Cisco Packet Tracer để mô phỏng một mạng LAN đơn giản gồm 1 Switch và 4 máy tính. Các máy tính sẽ được cấu hình địa chỉ IP tĩnh để kiểm tra kết nối bằng lệnh ping.

Bảng 5.1. Bảng cấu hình thiết bị

Thiết bị	Địa chỉ IP	Subnet Mask
PC1	192.168.1.1	255.255.255.0
PC2	192.168.1.2	255.255.255.0
PC3	192.168.1.3	255.255.255.0
Server	192.168.1.254	255.255.255.0
Switch	(Không cần IP)	—

Các bước thực hiện:

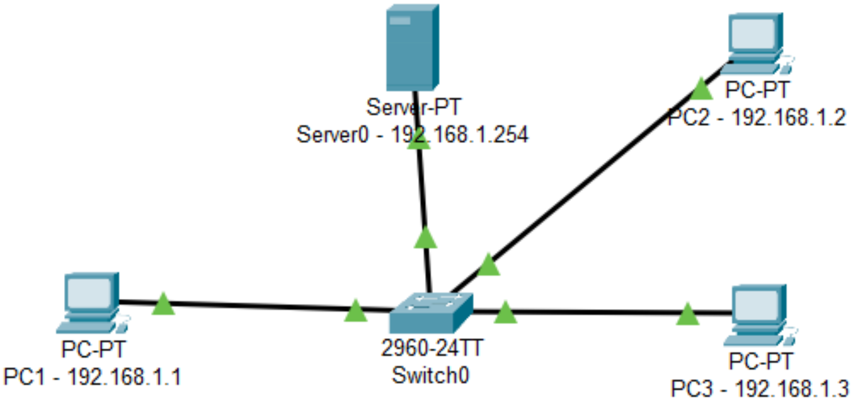
Bước 1: Mở phần mềm Packet Tracer.

Bước 2: Kéo thả các thiết bị từ thanh công cụ:

- 1 Switch (ví dụ: 2960).
- 3 PC.
- 1 Server

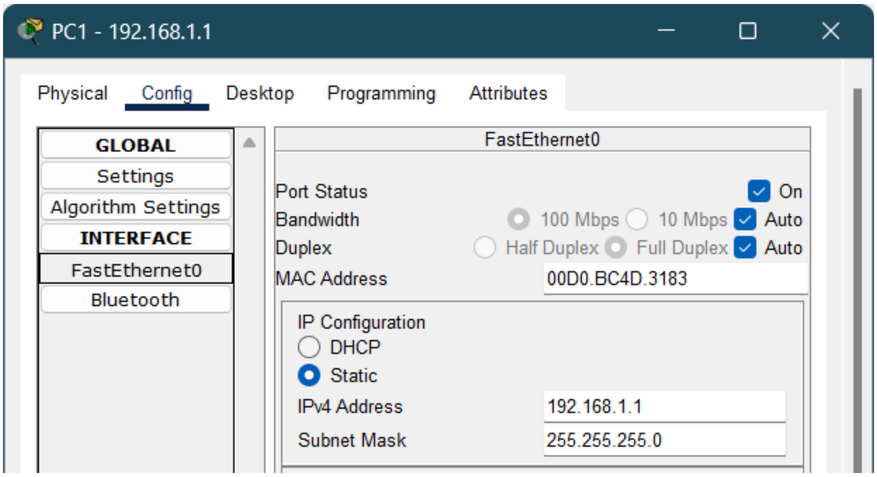
Bước 3: Dùng dây mạng Straight-Through để kết nối từng PC với switch.

Kết quả thu được mô hình mạng như Hình 5.18.


Server-PT
Server0 - 192.168.1.254
PC-PT
PC2 - 192.168.1.2
PC-PT
PC1 - 192.168.1.1
2960-24TT
Switch0
PC-PT
PC3 - 192.168.1.3

Hình 5.18. Mô hình mạng cần xây dựng

Bước 4: Nhấn vào từng PC → chọn tab Config → FastEthernet0 → IP Configuration. Nhập địa chỉ IP và Subnet Mask như bảng phân phối IP trên.


PC1 - 192.168.1.1
Physical
Config
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth

FastEthernet0

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

MAC Address 00D0.BC4D.3183

IP Configuration

○ DHCP

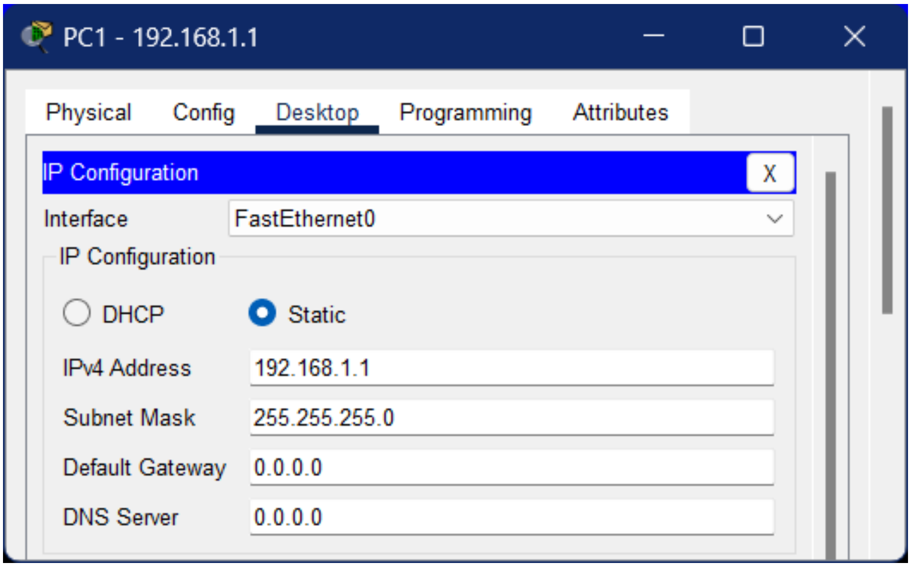
● Static

IPv4 Address 192.168.1.1

Subnet Mask 255.255.255.0

Hình 5.19. Cấu hình địa chỉ IP trên tab Config

Một cách khác để cấu hình địa chỉ IP là sử dụng tiện ích IP Configuration trong tab Desktop. Chọn Interface “FastEthernet0” và nhập thông tin cấu hình.


PC1 - 192.168.1.1
Physical
Config
Desktop
Programming
Attributes
X

IP Configuration

Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IPv4 Address: 192.168.1.1

Subnet Mask: 255.255.255.0

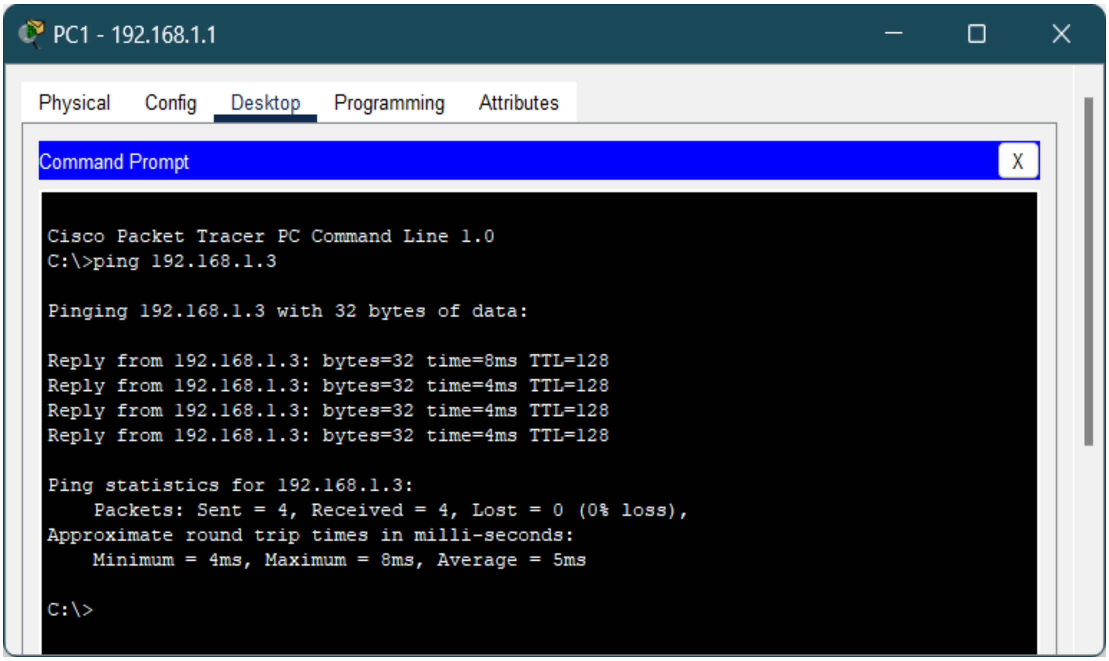
Default Gateway: 0.0.0.0

DNS Server: 0.0.0.0

Hình 5.20. Cấu hình địa chỉ IP bằng tiện ích IP Configuration

Bước 5: Kiểm tra kết nối:

- Mở Command Prompt trên PC1, gõ: ping 192.168.1.3


PC1 - 192.168.1.1
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X

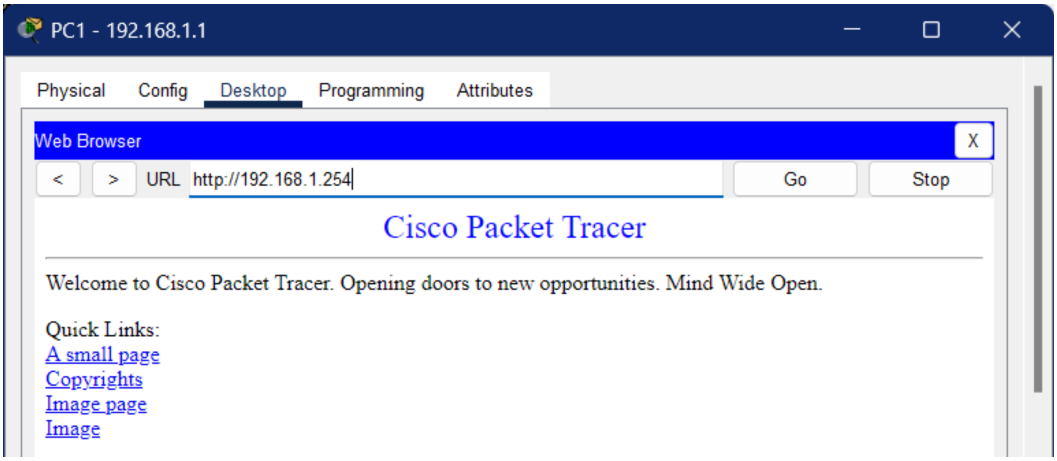
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes=32 time=8ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128
Reply from 192.168.1.3: bytes=32 time=4ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms

C:\>
```

Hình 5.21. Kết quả thực hiện lệnh ping trên PC1

- Thực hiện tương tự để kiểm tra với các máy còn lại.

Bước 6: Kiểm tra web server

Trên PC1 mở Web Browser và gõ vào địa chỉ 192.168.1.254 (địa chỉ IP của web server).



http://192.168.1.254

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

Hình 5.22. Kết quả chạy trình duyệt trên PC1

Nếu mạng được cấu hình đúng, trên chương trình trình duyệt của PC1 sẽ thể hiện một trang web đơn giản như Hình 5.22.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

Sử dụng phần mềm Packet Tracer vẽ sơ đồ mạng trong gia đình mình và cấu hình các thiết bị trong đó để có thể thực hiện truyền dữ liệu giữa các máy tính.

Câu hỏi lý thuyết

1) Nếu bạn thay thế một thiết bị Hub bằng Switch trong cùng một mạng nội bộ, điều gì sẽ thay đổi về cách các gói dữ liệu được xử lý và hiệu suất truyền dữ liệu?
2) Tại sao switch cần có bộ nhớ để lưu địa chỉ MAC, còn hub thì không? Hãy liên hệ với cách hoạt động thực tế của hai thiết bị này.
3) Giả sử trong một mạng có nhiều thiết bị gửi dữ liệu đồng thời, tại sao việc sử dụng Hub có thể gây ra hiện tượng va chạm gói tin (collision), còn Switch thì ít khi xảy ra điều này?
4) Router thường được dùng để kết nối giữa các mạng khác nhau. Vậy nếu bạn chỉ dùng Switch hoặc Hub trong một hệ thống nhiều mạng con, hệ thống sẽ gặp hạn chế gì?
5) Hãy suy luận: nếu một thiết bị hoạt động ở cấp độ địa chỉ MAC, thiết bị đó có thể định tuyến gói tin đến mạng khác không? Vì sao?
6) Trong một sơ đồ mạng nhỏ, nếu bạn thay thế một router bằng một switch, bạn sẽ mất đi khả năng nào? Điều này ảnh hưởng thế nào tới kết nối Internet?
7) Tại sao việc cấu hình địa chỉ IP lại thường liên quan tới router, mà không áp dụng cho switch hoặc hub? Hãy phân tích theo chức năng.

8) Bạn có thể kết nối hai mạng LAN bằng thiết bị nào? Có bắt buộc phải dùng router không? Trong trường hợp nào có thể thay thế bằng thiết bị khác?
9) Nếu thiết bị switch lưu sai bảng ánh xạ địa chỉ MAC, điều gì sẽ xảy ra với việc truyền dữ liệu trong mạng LAN?
10) Trong mạng LAN, bạn muốn ưu tiên truyền dữ liệu cho camera IP để giảm trễ. Thiết bị nào có thể hỗ trợ cấu hình ưu tiên như vậy? Vì sao hub không làm được điều đó?

BÀI 6. TRUYỀN THÔNG TIN TRONG MẠNG

Mục tiêu của bài học

Bài học giúp sinh viên hiểu rõ quá trình truyền dữ liệu trong mạng máy tính từ góc nhìn thực tiễn và lý thuyết. Nội dung bao gồm việc phân tích vai trò của địa chỉ IP và MAC, hoạt động của các giao thức như ARP, ICMP, định tuyến qua router, cũng như mô hình hóa các quá trình truyền thông qua các kiến trúc mạng.

Về kiến thức:

- Trình bày được vai trò và cách sử dụng của địa chỉ IPv4 trong truyền thông mạng.
- Mô tả hoạt động của giao thức ARP và phân biệt truyền thông trong mạng LAN và liên mạng.
- Hiểu rõ các khái niệm: unicast, broadcast, multicast, xung đột và miền quảng bá.
- Giải thích cấu trúc và chức năng của các mô hình mạng OSI và TCP/IP.

Về kỹ năng:

- Sử dụng Packet Tracer ở chế độ Simulation để mô phỏng quá trình truyền dữ liệu và phân tích các gói tin ARP, ICMP.
- Cấu hình địa chỉ IP, gateway, DNS cho các thiết bị đầu cuối trong mô hình mạng.
- Phân tích được quá trình đóng gói, định tuyến và truyền dữ liệu trong mô hình mạng thực tế.

6.1. Truyền thông tin trong mạng

6.1.1. Địa chỉ IPv4

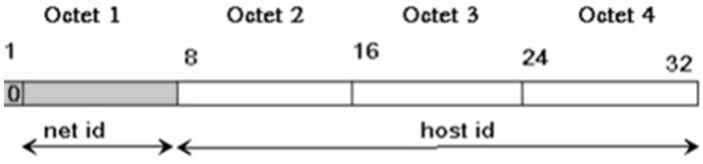
Trong bài học số 4 chúng ta đã học về cấu trúc của địa chỉ IPv4. Bài học số 5 chúng ta đã thực hiện một số thao tác cấu hình địa chỉ IP cho thiết bị trong bài thực hành mô phỏng mạng với Packet Tracer.

Trong bài học này chúng ta không nhắc lại kiến thức về cấu trúc địa chỉ IP mà tập trung vào khía cạnh sử dụng và quản lý địa chỉ IP trong hệ thống mạng. Việc hiểu khía cạnh quản lý và sử dụng địa chỉ IP có vai trò quan trọng khi xem xét quá trình truyền thông tin trong LAN và liên mạng.

a) Phân lớp địa chỉ IP

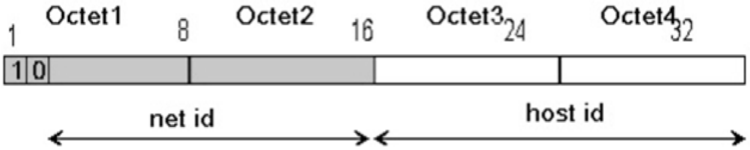
Không gian địa chỉ IPv4 được chia thành 5 lớp là : A, B, C, D, E. Việc phân chia này dựa vào giá trị Octet đầu tiên trong địa chỉ IPv4.

Lớp A: bit đầu tiên của octet đầu tiên là 0, do đó giá trị của octet đầu tiên là từ 0-127.



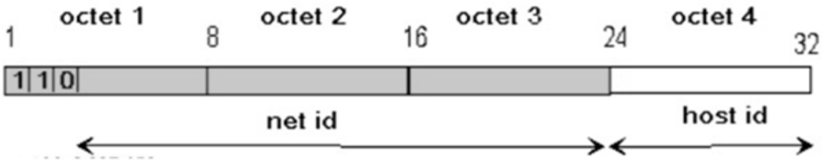
- Sử dụng một octet đầu làm phần mạng, ba octet sau làm phần host.
- Bit đầu của một địa chỉ lớp A luôn được giữ là 0.
- Các địa chỉ mạng lớp A gồm: 1.0.0.0 -> 126.0.0.0.
- Mạng 127.0.0.0 được sử dụng làm mạng loopback.
- Phần host có 24 bit => mỗi mạng lớp A có (2^24-2) host.

Lớp B: hai bit đầu tiên luôn là 1 0, và sử dụng 2 octet đầu cho phần mạng.



- Giá trị của Octet đầu tiên là 128-191
- Địa chỉ lớp B sử dụng hai octet đầu làm phần mạng, hai octet sau làm phần host.
- Hai bit đầu của một địa chỉ lớp B luôn được giữ là 1 0.
- Các địa chỉ mạng lớp B gồm: 128.0.0.0 -> 191.255.0.0. Có tất cả 2^14 mạng trong lớp B.
- Phần host dài 16 bit do đó một mạng lớp B có (2^16-2) host.

Lớp C: ba bit đầu tiên luôn là 110. Do đó giá trị của Octet đầu tiên là 192-223



- Địa chỉ lớp C sử dụng ba octet đầu làm phần mạng, một octet sau làm phần host.
- Ba bit đầu của một địa chỉ lớp C luôn được giữ là 1 1 0.
- Các địa chỉ mạng lớp C gồm: 192.0.0.0 -> 223.255.255.0. Có tất cả 2^21 mạng trong lớp C.
- Phần host dài 8 bit do đó một mạng lớp C có (2^8 - 2) host.

Lớp D: Giá trị của Octet đầu tiên là: 224-239

- Gồm các địa chỉ thuộc dải: 224.0.0.0 -> 239.255.255.255
- Được sử dụng làm địa chỉ multicast.

Ví dụ: 224.0.0.5 dùng cho OSPF; 224.0.0.9 dùng cho RIPv2

Lớp E: Giá trị của Octet đầu tiên là: 240-255. Được sử dụng trong môi trường Experimental.

Trong 5 lớp địa chỉ trên thì chỉ dùng 3 lớp địa chỉ A, B, C để cấp cho các host ở trong mạng.

b) Các địa chỉ IP đặc biệt

Nếu một địa chỉ IP có các bit thuộc phần định danh host đồng thời bằng 0 thì được gọi là **địa chỉ mạng** (network IP). Ví dụ: 192.168.10.0/24 là một *địa chỉ mạng*. Địa chỉ mạng là địa chỉ đại diện cho toàn bộ mạng, thay vì là địa chỉ của một host cụ thể. Vì vậy, địa chỉ mạng không được phép gán cho host.

Nếu một địa chỉ IP có tất cả các bit thuộc phần định danh đều bằng 1 thì được gọi là **địa chỉ quảng bá** (broadcast IP). Địa chỉ quảng bá là loại địa chỉ IP được sử dụng để gửi dữ liệu đồng thời tới tất cả các host trong mạng. Do đó, địa chỉ quảng bá cũng không được phép gán cho host. Ví dụ: 192.168.10.255/24 là một địa chỉ quảng bá cho địa chỉ mạng 192.168.10.0/24.

Địa chỉ 127.x.x.x có tên gọi riêng là **địa chỉ loopback** hoặc localhost. Trên máy tính thường sử dụng nhất là địa chỉ 127.0.0.1. Địa chỉ loopback được sử dụng khi cần khai thác dịch vụ truyền thông nhưng không muốn đưa dữ liệu lên mạng. Ví dụ, khi cả chương trình client và server cùng cài đặt trên cùng một thiết bị đầu cuối thì nên sử dụng địa chỉ loopback để truy cập server.

Các địa chỉ IP trong dải từ 169.254.0.0 - 169.254. 255.255 được gọi là **địa chỉ IP riêng tự động** (Automatic Private IP Address). Đây là địa chỉ IP được hệ điều hành tự động chọn ngẫu nhiên nếu như cơ chế nhận IP tự động (cơ chế DHCP) không đem về kết quả.

c) Địa chỉ public và địa chỉ private

Địa chỉ IPv4 được phân thành 2 loại:

IP private: chỉ được sử dụng trong mạng nội bộ (mạng LAN). Có thể được sử dụng lặp lại trong các mạng LAN khác nhau. Router sẽ chặn hết các gói tin chứa địa chỉ IP private không cho gói tin ra Internet. Dải địa chỉ IP private được quy định trong RFC 1918 như sau:

- Lớp A: 10.x.x.x (từ 10.0.0.0 đến 10.255.255.255)
- Lớp B: 172.16.x.x đến 172.31.x.x (từ 172.16.0.0 đến 172.31.255.255)
- Lớp C: 192.168.x.x (từ 192.168.0.0 đến 192.168.255.255)

IP public: là địa chỉ được sử dụng cho các gói tin đi trên môi trường Internet, được định tuyến trên môi trường Internet. Địa chỉ public phải là duy nhất cho mỗi host tham gia vào Internet.

Để kiểm tra địa chỉ IP public của một máy tính đang kết nối Internet, truy cập vào website https://whatismyipaddress.com/

d) Quản lý địa chỉ IP

Do địa chỉ IP là một tài nguyên sử dụng chung rất quan trọng trong hệ thống mạng, kho địa chỉ IP được quản lý một cách thống nhất bởi IANA - Internet Assigned Numbers

Authority. IANA là cơ quan thuộc ICANN (Internet Corporation for Assigned Names and Numbers) - một tổ chức phi lợi nhuận có trụ sở tại Hoa Kỳ. Vai trò chính của IANA bao gồm:

- Quản lý không gian địa chỉ IP (IPv4 và IPv6).
- Quản lý tên miền cấp cao (Top-Level Domains - TLDs).
- Phân bổ các số hiệu giao thức (protocol numbers) và cổng (port numbers).

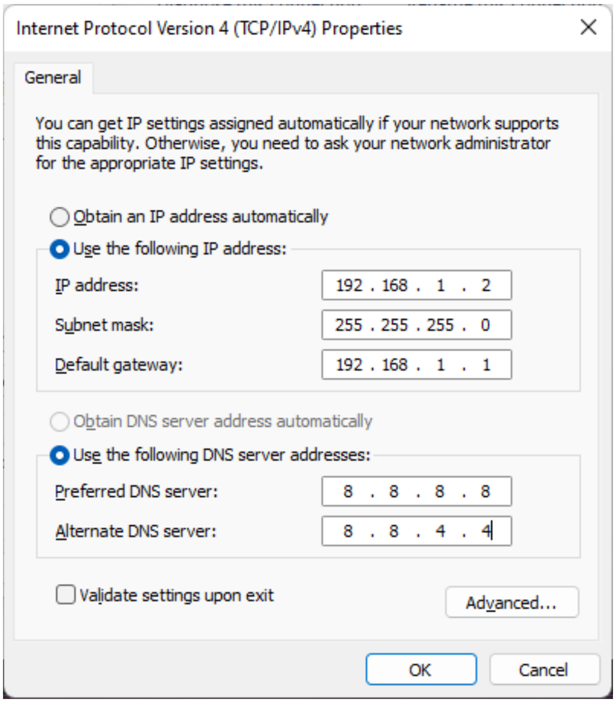
IANA không cấp địa chỉ IP trực tiếp cho từng tổ chức hay cá nhân, mà phân bổ cho 5 tổ chức quản lý cấp khu vực (RIR - Regional Internet Registries):

- ARIN - Bắc Mỹ (American Registry for Internet Numbers)
- RIPE NCC - Châu Âu, Trung Đông, Trung Á
- APNIC - Châu Á - Thái Bình Dương
- LACNIC - Mỹ Latin và vùng Caribbean
- AFRINIC - Châu Phi

Các RIR này tiếp tục phân bổ IP cho các nhà cung cấp dịch vụ Internet (ISP), các doanh nghiệp lớn, các tổ chức chính phủ, học thuật...

Ở Việt Nam, kho địa chỉ IP được quản lý bởi VNNIC - Trung tâm Internet Việt Nam, thuộc Bộ Thông tin và Truyền thông. VNNIC nhận cấp phát IP từ APNIC, và phân bổ cho các ISP và tổ chức trong nước.

e) Cấu hình TCP/IPv4



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 1 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 8 . 8 . 8 . 8

Alternate DNS server: 8 . 8 . 4 . 4

☐ Validate settings upon exit

Hình 6.1. Cấu hình địa chỉ IP trên Windows

Để diễn ra đầy đủ quá trình truyền thông tin, trên thiết bị đầu cuối phải cấu hình đầy đủ và chính xác các tham số IP để giao thức này có thể hoạt động.

Trên các hệ điều hành khác nhau, cách cấu hình có sự khác biệt lớn. Tuy nhiên, tên gọi và ý nghĩa các tham số về cơ bản là giống nhau.

Trên hệ điều hành windows, việc cấu hình cho giao thức IP được thể hiện trong Control Panel => Network and Internet => Network Connections.

Trong đó:

- IP address và Subnet mask đã được trình bày ở nội dung về địa chỉ IP.
- Default gateway là địa chỉ IP của cổng router kết nối với mạng cục bộ của máy tính. Vai trò của default gateway sẽ được nói tới trong phần tiếp theo của bài này.
- Preferred và Alternate DNS server là địa chỉ IP của các máy chủ DNS, nơi chịu trách nhiệm chuyển đổi từ tên miền về địa chỉ IP.

Cấu hình IP sẽ đi theo từng giao diện mạng cụ thể và phù hợp với mạng. Ví dụ, nếu thiết bị đầu cuối có 2 card mạng, mỗi card kết nối vào một mạng khác nhau thì cần cấu hình IP cho cả 2 giao diện này theo dải địa chỉ IP của mạng tương ứng.

6.1.2. Giao thức ARP

Trong một mạng TCP/IP, mỗi thiết bị đầu cuối đều được gán hai loại địa chỉ: địa chỉ IP (logic) và địa chỉ MAC (vật lý). Địa chỉ IP hỗ trợ định tuyến dữ liệu giữa các mạng khác nhau, trong khi địa chỉ MAC được sử dụng để truyền dữ liệu trong cùng một mạng LAN (miền quảng bá).

Khi hai thiết bị trong cùng một mạng LAN cần giao tiếp với nhau, việc xác định địa chỉ MAC của thiết bị đích là bắt buộc để đóng gói và gửi frame Ethernet. Để thực hiện điều này, giao thức ARP (Address Resolution Protocol) được sử dụng. ARP là một giao thức thiết yếu trong bộ giao thức TCP/IP, có chức năng ánh xạ địa chỉ IP sang địa chỉ MAC tương ứng.

Minh họa quá trình hoạt động của ARP:

- Giả sử máy tính A (IP: 192.168.1.2/24) muốn gửi dữ liệu đến máy tính B (IP: 192.168.1.3/24) trong cùng một mạng LAN:
- Kiểm tra địa chỉ IP đích: Máy A xác định rằng địa chỉ IP đích cùng mạng với địa chỉ IP của nó.
- Gửi yêu cầu ARP: Máy A phát một gói ARP dưới dạng quảng bá để hỏi “Ai là 192.168.1.3? Hãy gửi cho tôi địa chỉ MAC của bạn.”
- Nhận phản hồi ARP: Nếu máy B đang hoạt động, nó sẽ trả lời bằng cách gửi địa chỉ MAC của mình trở lại cho máy A.
- Gửi frame dữ liệu: Sau khi biết địa chỉ MAC của B, máy A có thể đóng gói frame Ethernet và gửi đến máy B.

- Ngược lại, nếu máy B không phản hồi (do tắt máy hoặc không kết nối mạng), máy A sẽ không thể tạo frame, và quá trình truyền dữ liệu thất bại.

Trường hợp liên mạng:

Nếu máy A có IP là 192.168.1.2 và muốn gửi dữ liệu đến địa chỉ 10.10.10.10 (khác mạng), thì máy A sẽ xác định rằng IP đích không cùng mạng. Lúc này, nó sẽ gửi yêu cầu ARP để tìm địa chỉ MAC của gateway mặc định (default gateway) và chuyển frame đến router, thay vì gửi trực tiếp đến máy B, ngay cả khi máy B nằm gần về mặt vật lý.

Như vậy, trong mạng TCP/IP, cả địa chỉ IP và MAC đều cần thiết để truyền dữ liệu. ARP là cầu nối giữa hai loại địa chỉ này trong quá trình giao tiếp trong mạng LAN.

6.1.3. Truyền thông tin trong mạng

a) Truyền thông tin trong LAN

Truyền thông trong mạng LAN dựa trên khả năng định vị địa chỉ MAC của thiết bị đích. Quá trình truyền thông diễn ra như sau:

- **Nếu thiết bị đích cùng mạng với thiết bị nguồn:**
    - Thiết bị nguồn kiểm tra bảng MAC (MAC Table) để tìm địa chỉ MAC tương ứng với IP đích.
    - Nếu địa chỉ MAC chưa có trong bảng, thiết bị sử dụng giao thức ARP để truy vấn địa chỉ MAC.
    - Sau khi nhận được phản hồi từ ARP, địa chỉ MAC sẽ được lưu vào MAC Table để sử dụng cho các lần truyền sau.
    - Thiết bị nguồn sử dụng địa chỉ MAC đã biết để đóng gói frame và gửi đến đích.
- **Nếu thiết bị đích thuộc mạng khác (khác dải địa chỉ IP):**
    - Thiết bị nguồn kiểm tra xem có khai báo gateway mặc định hay không.
    - Nếu có gateway mặc định, frame sẽ được gửi đến địa chỉ MAC của router (default gateway).
    - Nếu không có gateway, thiết bị không thể chuyển tiếp gói tin, và hệ điều hành sẽ thông báo lỗi: **"Destination host unreachable"**.

b) Truyền thông tin liên mạng

Khi hai thiết bị nằm ở các mạng IP khác nhau, việc truyền dữ liệu cần có sự hỗ trợ của các thiết bị định tuyến (router). Quá trình truyền thông tin liên mạng diễn ra như sau:

1. **Tại máy nguồn:**
    - Gói dữ liệu được đóng gói trong packet IP.

  - Packet IP tiếp tục được đóng gói vào frame Ethernet với địa chỉ MAC đích là địa chỉ của gateway mặc định.

2. **Tại router:**
   - Router nhận frame và giải nén để lấy packet IP.
   - Router sử dụng địa chỉ IP đích trong packet để tra bảng định tuyến nhằm xác định hướng chuyển tiếp phù hợp.
   - Nếu mạng đích kết nối trực tiếp với router, router sẽ đóng gói lại packet vào một frame mới và chuyển tiếp đến mạng đích.
   - Nếu không, packet được chuyển qua các router trung gian cho đến khi đến router có mạng đích kết nối trực tiếp.
3. **Quy tắc xử lý đặc biệt:**
   - **Địa chỉ broadcast:** Router sẽ chặn và không chuyển tiếp các packet có địa chỉ quảng bá (broadcast).
   - **Địa chỉ IP private:** Router cũng không chuyển tiếp packet chứa địa chỉ IP nằm trong dải địa chỉ riêng (private) ra ngoài mạng công cộng.

Lưu ý: Việc chuyển tiếp gói tin giữa các mạng phụ thuộc hoàn toàn vào **bảng định tuyến** được thiết lập tại router. Bảng định tuyến có thể được cấu hình tĩnh (thủ công) hoặc cập nhật động bằng các giao thức định tuyến như RIP, OSPF, BGP...

6.1.4. Các khái niệm liên quan đến truyền thông tin

a) Các chế độ truyền thông tin

Chế độ truyền thông tin (Transmission Mode, Communication Mode, Directional Mode) thể hiện chiều truyền thông tin giữa hai thiết bị trong mạng. Chế độ truyền thông tin có thể được định nghĩa ở cấp độ tín hiệu hoặc ở cấp độ dữ liệu.

Người ta phân biệt ba chế độ truyền thông tin:

1. Chế độ **đơn công** (simplex/simple mode): là chế độ truyền thông tin chỉ theo một chiều, hoặc gửi, hoặc nhận, không thể thực hiện cả hai. Một thiết bị chỉ có thể chuyên gửi ***hoặc*** nhận thông tin. Thông tin chỉ có thể di chuyển theo một chiều.
2. Chế độ **bán song công** (half-duplex mode): là chế độ truyền thông tin hai chiều không đồng thời. Một thiết bị có thể gửi ***và*** nhận dữ liệu, nhưng không phải tại cùng một thời điểm. Thông tin có thể di chuyển theo hai chiều nhưng không đồng thời.
3. Chế độ **song công** (full-duplex mode): là chế độ truyền thông tin hai chiều đồng thời. Một thiết bị có thể nhận ***và*** gửi dữ liệu tại ***cùng một thời điểm***. Thông tin có thể đồng thời di chuyển theo cả hai chiều.

b) Các phương thức truyền thông tin

Phương thức truyền thông tin (transmission method) thể hiện số lượng thiết bị có thể tham gia vào quá trình truyền thông tin.

Trong mạng máy tính có thể phát sinh nhu cầu truyền thông tin từ một thiết bị tới một, nhiều, hoặc tất cả các thiết bị khác. Do đó có các phương thức truyền thông tin khác nhau.

1. Nếu một thiết bị truyền thông tin tới đúng một thiết bị khác, nó được gọi là **truyền thông unicast**. Đây là mô hình truyền thông tin cơ bản nhất và được sử dụng rộng rãi nhất. Mọi loại mạng đều hỗ trợ mô hình này.
2. Nếu một thiết bị truyền thông tin tới tất cả các thiết bị khác trong mạng, nó được gọi là **truyền thông broadcast** (truyền thông quảng bá). Truyền thông quảng bá rất quan trọng đối với hoạt động của một số giao thức mạng làm nhiệm vụ hỗ trợ quá trình truyền thông. Vùng của mạng nơi có thể diễn ra truyền thông broadcast cũng được gọi là **miền quảng bá** (broadcast domain).
3. Nếu một thiết bị truyền thông tin tới một nhóm thiết bị khác, nó được gọi là **truyền thông multicast**. Truyền thông multicast thường được sử dụng bởi thiết bị router và các giao thức định tuyến để xác định tình trạng của mạng.

Các phương thức này diễn ra ở nhiều cấp độ, cơ bản nhất là cấp độ mạng cục bộ, và cấp độ liên mạng.

Ở cấp độ mạng cục bộ, các công nghệ mạng LAN hiện nay hỗ trợ đầy đủ cả 3 mô hình. Ở cấp độ liên mạng chủ yếu cho phép truyền thông unicast, không cho phép truyền thông broadcast, cũng như hạn chế truyền thông multicast.

Trong mạng LAN, để thực hiện truyền thông broadcast, frame dữ liệu cần sử dụng một địa chỉ MAC đích đặc biệt với tất cả các bit bằng 1. Thông thường, các công nghệ LAN không phân biệt giữa truyền thông multicast và broadcast.

c) Xung đột và miền xung đột

Khi tìm hiểu về cách thức hoạt động của thiết bị hub, chúng ta có thể thấy, loại thiết bị này cho phép mọi thiết bị đầu cuối gắn với nó tự do truyền tín hiệu tới bất kì đích nào. Cả mạng khi này như một môi trường truyền dẫn được dùng chung bởi nhiều thiết bị đầu cuối. Nói một cách khác, môi trường mạng lúc này được chia sẻ bởi nhiều thiết bị đầu cuối với nhau.

Khi một môi trường truyền được chia sẻ bởi nhiều thiết bị có thể dẫn đến hiện tượng xung đột tín hiệu.

Xung đột tín hiệu là tình trạng tín hiệu đến từ nhiều nguồn "va chạm" nhau ở một điểm nào đó trên đường truyền làm triệt tiêu hoặc cộng dồn tín hiệu. Kết quả của xung đột là tín hiệu bị biến dạng và trở nên vô nghĩa. Khi xảy ra xung đột, quá trình truyền thông tin được xem là thất bại.

Vùng mạng nơi tín hiệu được phép di chuyển tự do (không có sự cản trở của thiết bị mạng) và có khả năng xảy ra xung đột tín hiệu được gọi là **miền xung đột** (collision domain). Miền xung đột càng rộng, khả năng xung đột càng lớn. Miền xung đột nhỏ ít khả năng gây xung đột hơn. Do đó, miền xung đột càng nhỏ càng tốt.

Thiết bị hub cho phép ghép nối thêm nhiều thiết bị trong khi vẫn cho phép tín hiệu truyền tự do sẽ làm *mở rộng miền xung đột*. Hậu quả là hiệu suất truyền thông của mạng sẽ giảm.

Thiết bị switch không làm mở rộng miền xung đột. Thay vào đó, chúng *tách một miền xung đột lớn thành nhiều miền xung đột nhỏ*. Do đó, khả năng xung đột dữ liệu giảm và giúp tăng hiệu suất truyền thông.

d) Miền quảng bá

Vùng của mạng nơi frame dữ liệu được phép di chuyển tự do và không bị ngăn chặn bởi thiết bị mạng được gọi là một **miền quảng bá** (Broadcast Domain).

Miền quảng bá quan trọng ở chỗ, nó là "địa bàn" hoạt động của các giao thức mạng LAN. Chỉ các thiết bị đầu cuối ở trong cùng một miền quảng bá mới có thể gửi frame dữ liệu cho nhau. Có thể hình dung rằng, mỗi miền quảng bá tương đương với một mạng LAN cơ bản (mạng LAN không chứa router).

Các thiết bị mạng như hub và switch đều có tác dụng mở rộng miền quảng bá, tức là mở rộng mạng LAN. Thiết bị router lại có tác dụng ngăn chặn frame quảng bá, và qua đó nó có tác dụng chia nhỏ miền quảng bá.

Theo cách hiểu như trên, cũng như qua nguyên tắc hoạt động của thiết bị, có thể nói rằng một miền xung đột sẽ thuộc về một miền quảng bá, và mỗi miền quảng bá có thể chứa nhiều miền xung đột.

6.2. Thực hành - mô phỏng quá trình truyền thông tin sử dụng Packet Tracer

Mục tiêu:

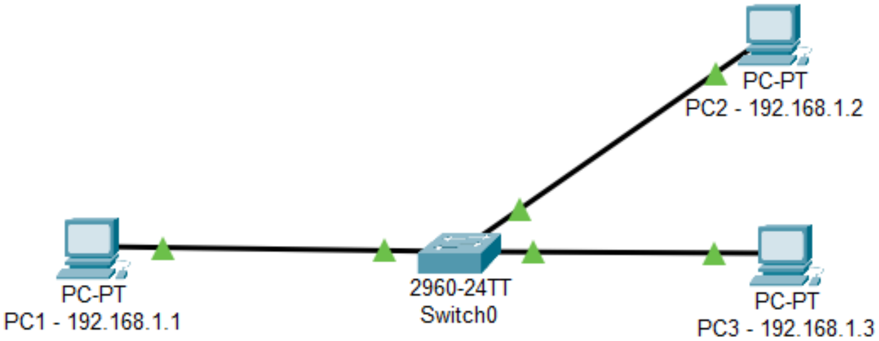
- Giúp sinh viên hình dung một cách trực quan đường đi của dữ liệu trong mạng LAN hình sao sử dụng switch;
- Nắm được cách sử dụng chế độ Simulation của Packet Tracer.

Để đạt được mục tiêu này, sinh viên sẽ xây dựng một mô hình mạng đơn giản với 3 PC và 1 switch. Ở chế độ Simulation, từ một PC sẽ ping tới một PC khác. Packet Tracer sẽ thể hiện đường đi của các gói tin cũng như việc xử lý ở các PC có liên quan. Chúng ta sẽ giới hạn chỉ theo dõi các gói tin ARP và ICMP để có hình dung trực quan về cách ARP và ICMP hoạt động trong mạng.

Chuẩn bị:

- Xây dựng một mạng đơn giản với một switch với 3 thiết bị đầu cuối;

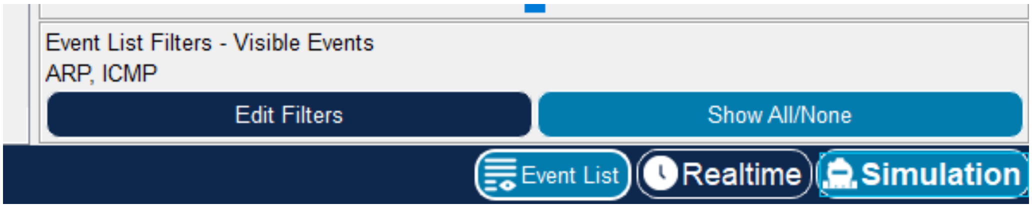
- Lần lượt cấu hình IP cho các NIC trên PC;
- Kiểm tra hoạt động của mạng bằng cách từ PC 1 ping tới PC 3.


PC-PT
PC2 - 192.168.1.2
PC-PT
PC1 - 192.168.1.1
2960-24TT
Switch0
PC-PT
PC3 - 192.168.1.3

Hình 6.2. Sơ đồ mạng mô phỏng

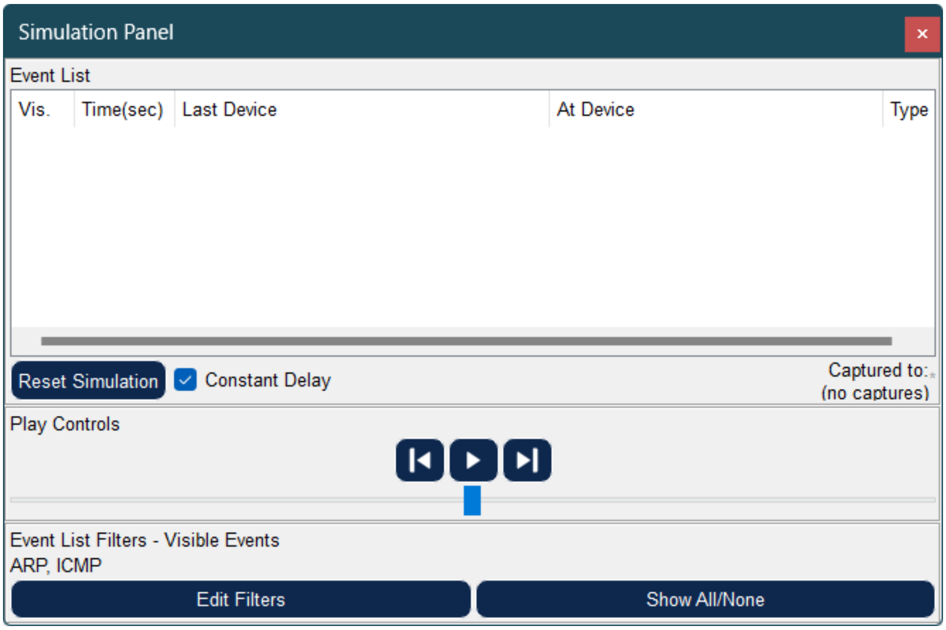
Thực hiện:

Bước 1: Chuyển Packet Tracer sang chế độ Simulation


Event List Filters - Visible Events
ARP, ICMP
Edit Filters
Show All/None
Event List
Realtime
Simulation

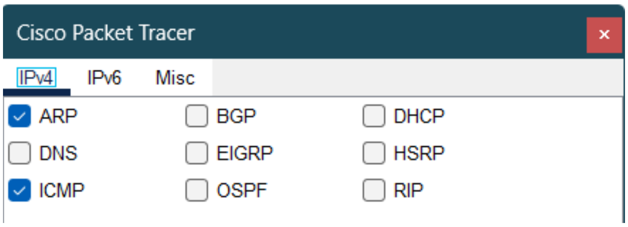
Hình 6.3. Nút chuyển chế độ Simulation trên Packet Tracer

Bước 2: Mở cửa sổ filter bằng cách bấm nút Edit Filters trên Simulation Panel.


Simulation Panel
×
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Reset Simulation
Constant Delay
Captured to:
(no captures)
Play Controls
Event List Filters - Visible Events
ARP, ICMP
Edit Filters
Show All/None

Hình 6.4. Giao diện Simulation Panel

Bước 3: Bỏ tích chọn của tất cả các mục trên cả 3 tab (IPv4, IPv6, Misc), chỉ để lại ARP và ICMP. Sau đó bấm nút play trên Simulation Panel để bắt đầu quá trình mô phỏng.


Cisco Packet Tracer
×
IPv4
IPv6
Misc

☑ ARP
☐ DNS
☑ ICMP
☐ BGP
☐ EIGRP
☐ OSPF
☐ DHCP
☐ HSRP
☐ RIP

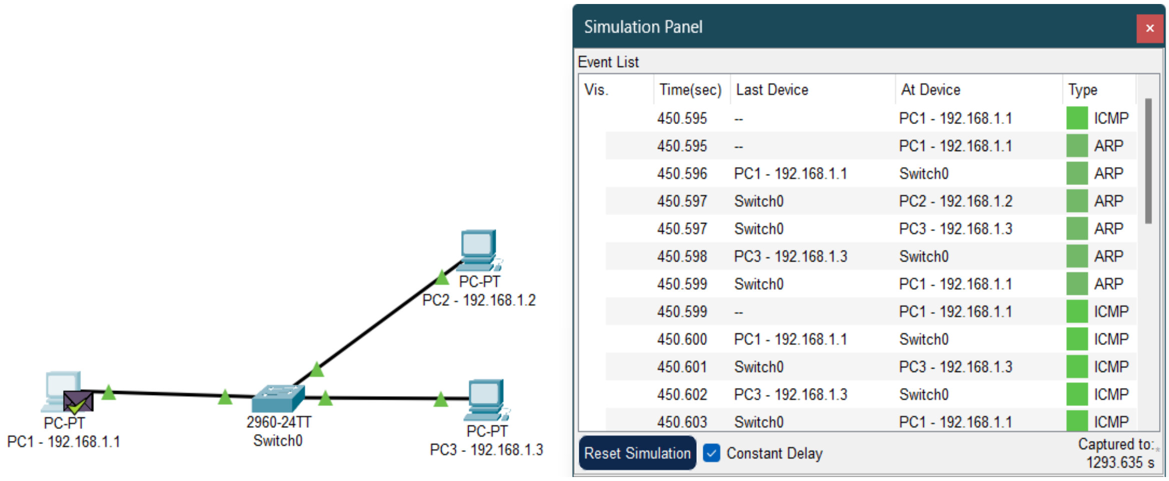
Hình 6.5. Các bộ lọc được sử dụng

Bước 4: Mở ứng dụng Command Prompt trên PC1 và chạy các lệnh sau:

```
arp - d
ping 192.168.1.3
```

Lệnh thứ nhất có tác dụng xóa bảng ARP trên máy cục bộ. Lệnh thứ hai sẽ gửi 4 gói tin ICMP echo request tới PC3.

Thu nhỏ cửa sổ Command Prompt để quan sát quá trình mô phỏng trên cửa sổ chính và Simulation Panel.


PC-PT
PC2 - 192.168.1.2
PC-PT
PC1 - 192.168.1.1
2960-24TT
Switch0
PC-PT
PC3 - 192.168.1.3
Simulation Panel
×
Event List
Reset Simulation
Captured to:
1293.635 s

Vis.	Time(sec)	Last Device	At Device	Type
	450.595	--	PC1 - 192.168.1.1	🟩 ICMP
	450.595	--	PC1 - 192.168.1.1	🟩 ARP
	450.596	PC1 - 192.168.1.1	Switch0	🟩 ARP
	450.597	Switch0	PC2 - 192.168.1.2	🟩 ARP
	450.597	Switch0	PC3 - 192.168.1.3	🟩 ARP
	450.598	PC3 - 192.168.1.3	Switch0	🟩 ARP
	450.599	Switch0	PC1 - 192.168.1.1	🟩 ARP
	450.599	--	PC1 - 192.168.1.1	🟩 ICMP
	450.600	PC1 - 192.168.1.1	Switch0	🟩 ICMP
	450.601	Switch0	PC3 - 192.168.1.3	🟩 ICMP
	450.602	PC3 - 192.168.1.3	Switch0	🟩 ICMP
	450.603	Switch0	PC1 - 192.168.1.1	🟩 ICMP

☑ Constant Delay

Hình 6.6. Kết quả mô phỏng trong Packet Tracer

Trong cửa sổ làm việc chính, Packet Tracer thể hiện gói dữ liệu theo hình một bức thư, và quá trình truyền gói dữ liệu là việc di chuyển bức thư đó từ node này đến node khác. Bức thư dừng ở node nào có nghĩa là đang xảy ra quá trình xử lý gói tin tại node đó.

Trong cửa sổ Simulation Panel, Packet Tracer thể hiện thông tin chi tiết về loại gói tin (cột Type), thiết bị nơi thực hiện việc xử lý (cột At Device), và đích tới nếu có (cột Last Device).

Kết quả mô phỏng có thể diễn giải lại như sau:

- PC1 tạo ra gói tin ICMP và đưa xuống đóng gói frame. Tuy nhiên, để đóng gói frame gửi tới PC3, nó phải có địa chỉ MAC của PC3. Bảng ARP hiện đang trống (do chúng ta dùng lệnh arp -d để xóa bảng ARP). Do đó PC1 tạm dừng đóng gói frame cho ICMP.
- PC1 sử dụng ARP để xác định địa chỉ MAC của PC3. Do đó, PC1 phát quảng bá một gói tin ARP. Gói tin ARP quảng bá này đi tới switch rồi tới PC2 và PC3.
- PC2 từ chối xử lý gói ARP.
- PC3 phản hồi trở lại PC1 với thông tin địa chỉ MAC.
- PC1 nhận địa chỉ MAC của PC3 và tiếp tục đóng gói frame chứa ICMP.
- Frame chứa gói ICMP gửi tới switch.
- Switch chuyển tiếp frame đó tới PC3.
- PC3 xử lý gói tin ICMP và gửi trở lại tới switch.
- Switch chuyển tiếp gói tin ICMP trở lại PC1. Đến đây là kết thúc cho gói tin ICMP đầu tiên.
- PC1 tạo ra gói tin ICMP thứ 2. Gói tin ICMP này ngay lập tức được đóng gói frame và di chuyển qua switch tới PC3. PC3 xử lý và gửi lại phản hồi trở lại cho PC1. Gói tin ICMP thứ 2 có thể tạo ra ngay lập tức mà không cần đến ARP do địa chỉ MAC của PC3 đã được lưu vào bảng ARP của PC1.
- Chương trình ping trên PC1 mặc định tạo ra và gửi đi 4 gói truy vấn ICMP. Packet Tracer thể hiện mỗi gói tin ICMP với một màu sắc riêng có thể quan sát trên Simulation Panel.

6.3. Mô hình mạng

Như các phần trên chúng ta đã thấy, trong một mạng máy tính có rất nhiều thành phần cả cứng và mềm cùng hoạt động. Chúng thực hiện những chức năng khác nhau nhưng tương tác với nhau để cùng thực hiện mục tiêu chung là truyền thông tin giữa các thiết bị.

Để phục vụ nghiên cứu, thiết kế và quản lý các thành phần của mạng, người ta xây dựng ra các mô hình mạng.

Mô hình mạng mô tả kiến trúc, thành phần và thiết kế được sử dụng để thiết lập giao tiếp giữa hệ thống nguồn và hệ thống đích.

Có thể hiểu một cách đơn giản rằng, mô hình mạng là một hệ thống phân loại các thành phần trong một mạng máy tính theo chức năng. Qua đó, việc học tập và nghiên cứu về mạng sẽ dễ dàng hơn. Khi nghiên cứu về mạng máy tính, mô hình mạng là một nội dung không thể thiếu. Nó cung cấp một cách nhìn hệ thống, đầy đủ, hoàn chỉnh về tất cả các thành phần của mạng máy tinh.

Có hai mô hình mạng được sử dụng phổ biến nhất là **mô hình kết nối các hệ thống mở** (mô hình OSI - Open System Interconnection) và **mô hình TCP/IP**. Ngoài ra một số mô hình khác cũng được tạo ra phục vụ cho nghiên cứu và giảng dạy về mạng máy tính.

6.3.1. Kiến trúc phân tầng

Tất cả các mô hình mạng đều xây dựng theo *kiến trúc phân tầng*. **Tầng** (layer) là sự phân chia logic về mặt chức năng. Có thể hình dung rằng trong mạng máy tính có rất nhiều thành phần, và mỗi thành phần đảm nhiệm một số chức năng cụ thể. Các thành phần này có thể được phân loại về mặt chức năng. Các thành phần với chức năng nhất định sẽ được phân loại về một tầng xác định.

Mỗi mô hình mạng bao gồm nhiều tầng xếp chồng lên nhau, trong đó:

- Số lượng, vai trò và chức năng của mỗi tầng được định nghĩa từ trước trong một mô hình mạng. Trong các mô hình mạng khác nhau, số tầng, tên của các tầng, và chức năng của các tầng cũng không giống nhau.
- Mỗi tầng chứa các thực thể. **Thực thể** có thể hiểu là phần mềm (thực thi giao thức) hoặc phần cứng, có nhiệm vụ xác định riêng biệt trong hệ thống mạng.
- Các thực thể ở các tầng liền kề mới tương tác được với nhau. Các thể ở các tầng không liền kề thì không tương tác với nhau. Các thực thể tương tác gián tiếp qua **giao diện tầng**. Có thể hình dung giao diện tầng như một dạng API - giao diện lập trình ứng dụng.
- Mỗi thực thể cung cấp dịch vụ cho thực thể tầng liền trên và khai thác dịch vụ của thực thể tầng liền dưới.
- Các thực thể ở cùng tầng có những điểm chung về dịch vụ nó cung cấp, mặc dù chúng có thể thực hiện theo những cách khác nhau. Vì vậy, khi nói về tầng nào, người ta có thể hình dung ra đặc điểm chung của tầng đó mà không cần nói tới các thực thể cụ thể ở tầng đó.

Để dễ hình dung, có thể nhìn nhận thực thể ở mỗi tầng giống như một phần mềm. Phần mềm ở tầng liền trên có thể gọi phần mềm ở tầng liền dưới thông qua API được định nghĩa sẵn. Nói cách khác, thực thể ở mỗi tầng hoạt động tương tự như một kiểu máy ảo cung cấp những dịch vụ nhất định cho tầng bên trên liền kề nó, nhưng không cho tầng trên biết dịch vụ đó được xây dựng ra sao.

Khi nhìn nhận các giao thức từ khía cạnh thực thi có thể hình dung rằng, các "chương trình" giao thức ở các tầng hoạt động gần giống như các hàm gọi lẫn nhau. Các hàm cấp độ cao hơn gọi các hàm ở cấp thấp hơn để thực hiện các nhiệm vụ cụ thể. Các hàm cấp dưới lại có thể gọi tới các hàm cấp dưới nữa.

Mô hình phân tầng này quy định rằng, chỉ có các hàm ở các cấp liền kề nhau mới có thể gọi nhau. Các hàm này gọi nhau theo kiểu "gọi qua giao diện", tức là không trực tiếp gọi mà thông qua một thành phần trung gian gọi là giao diện (interface).

Giao diện là một đơn vị chương trình đặc biệt chỉ chứa phần mô tả (bao gồm mô tả dữ liệu đầu vào và mô tả dữ liệu đầu ra) nhưng không chứa phần thực thi. Giao diện giúp một ứng dụng này sử dụng dịch vụ cung cấp bởi ứng dụng khác mà không phụ thuộc vào cách thực thi của dịch vụ đó. Cách thức làm việc này giúp các hàm không phụ thuộc chặt với nhau, khi cần thiết có thể thay thế hàm nọ bằng hàm kia mà không ảnh hưởng đến hàm cấp cao hơn.

6.3.2. Mô hình OSI

a) Tổng quan về mô hình OSI

Mô hình OSI (Open Systems Interconnection) là một *mô hình tham chiếu* do tổ chức ISO xây dựng nhằm mô tả cách thức hai hệ thống máy tính có thể giao tiếp với nhau thông qua một mạng truyền thông. Mô hình này phân chia quá trình truyền thông thành bảy tầng logic riêng biệt, mỗi tầng đảm nhiệm một nhóm chức năng cụ thể trong việc xử lý, truyền và quản lý dữ liệu qua mạng.

OSI được gọi là **mô hình tham chiếu** vì nó chỉ ra những gì cần có, chứ không chỉ rõ cách thực hiện ra sao. Nó đóng vai trò như một bản thiết kế lý thuyết để các kỹ sư mạng, nhà phát triển và nhà nghiên cứu tham khảo khi xây dựng các hệ thống truyền thông hiện đại.

Mục tiêu chính của mô hình OSI là chuẩn hóa các chức năng cần thiết để cho phép các hệ thống mạng khác nhau, dù sử dụng phần cứng hay phần mềm khác nhau, vẫn có thể kết nối và truyền thông với nhau một cách nhất quán và hiệu quả. Nhờ đó, các thiết bị và phần mềm từ các nhà cung cấp khác nhau vẫn có thể hoạt động tương thích trong cùng một mạng.

Điều quan trọng cần lưu ý: Mô hình OSI không phải là một giao thức cụ thể, mà là một khuôn khổ lý thuyết giúp các nhà phát triển thiết kế và xây dựng các giao thức mạng một cách có hệ thống và thống nhất.

Mô hình OSI bao gồm 7 tầng, theo thứ tự từ trên xuống dưới như sau: Tầng ứng dụng, tầng trình diễn, tầng phiên, tầng giao vận, tầng mạng, tầng liên kết dữ liệu, tầng vật lý (xem Bảng 6.2).

Bảng 6.1. Mô tả các tầng của mô hình OSI

Tầng	**Tên tầng**	**Mô tả ngắn**
7	Application (Ứng dụng)	Giao tiếp trực tiếp với phần mềm người dùng
6	Presentation (Trình diễn)	Xử lý định dạng dữ liệu (mã hóa, nén, giải mã…)
5	Session (Phiên)	Quản lý phiên giao tiếp giữa hai thiết bị
4	Transport (Giao vận)	Đảm bảo truyền dữ liệu từ đầu đến cuối
3	Network (Mạng)	Định tuyến và quản lý địa chỉ IP
2	Data Link (Liên kết dữ liệu)	Đảm bảo truyền dữ liệu tin cậy giữa các nút liền kề
1	Physical (Vật lý)	Truyền bit qua môi trường vật lý

Mô hình OSI đóng vai trò là một chuẩn khái niệm giúp các nhà phát triển phần mềm, kỹ sư mạng và sinh viên có một cái nhìn hệ thống và có tổ chức về các chức năng cần thiết trong truyền thông mạng. Dù ngày nay không phải tất cả các hệ thống mạng đều tuân theo mô hình OSI một cách chính xác (ví dụ như Internet sử dụng mô hình TCP/IP), nhưng mô hình OSI vẫn là cơ sở lý thuyết quan trọng và phổ biến trong giảng dạy và nghiên cứu mạng máy tính.

b) Chức năng của từng tầng trong mô hình OSI

Tầng ứng dụng (Application Layer): Là tầng gần người dùng nhất, cung cấp giao diện và dịch vụ mạng trực tiếp cho các ứng dụng như trình duyệt web, email, FTP... Các giao thức gắn với ứng dụng như HTTP (web), SMTP (email), FTP (truyền file) thực hiện nhiệm vụ phù hợp mô tả của tầng ứng dụng.

Tầng trình diễn (Presentation Layer): Đảm nhiệm việc định dạng, mã hóa và giải mã dữ liệu sao cho dữ liệu gửi đi có thể được hiểu đúng bởi phía nhận. Tầng này cũng đảm bảo dữ liệu có thể được mã hóa để bảo mật và nén để tối ưu dung lượng truyền tải.

Tầng phiên (Session Layer): Quản lý các phiên làm việc (session) giữa các ứng dụng, bao gồm thiết lập, duy trì và kết thúc phiên. Tầng này cho phép tổ chức trao đổi thông tin theo kiểu hội thoại, giúp đồng bộ hóa và quản lý giao tiếp giữa các tiến trình trên hai hệ thống khác nhau.

Tầng giao vận (Transport Layer): Cung cấp cơ chế truyền dữ liệu từ đầu cuối đến đầu cuối (dịch vụ end-to-end). Tầng này thực hiện phân mảnh và tái kết hợp dữ liệu, kiểm soát lỗi và luồng dữ liệu, đồng thời đảm bảo việc truyền tải không bị gián đoạn. Các giao TCP và UDP đã học thực hiện các nhiệm vụ phù hợp với mô tả của tầng này.

Tầng mạng (Network Layer): Đảm nhiệm việc định tuyến và chuyển tiếp gói tin trong mạng để thực hiện dịch vụ truyền thông host-to-host. Tầng này xác định địa chỉ logic (như địa chỉ IP) và thiết lập các đường đi tối ưu để truyền dữ liệu giữa các mạng khác nhau. Giao thức IP thực hiện nhiệm vụ phù hợp với mô tả của tầng này.

Tầng liên kết dữ liệu (Data Link Layer): Cung cấp phương thức truyền dữ liệu giữa hai nút trực tiếp trên cùng một đường truyền vật lý. Tầng này thực hiện việc đánh khung (framing), phát hiện và sửa lỗi, điều khiển truy cập đường truyền (MAC - Media Access Control).

Tầng vật lý (Physical Layer): Là tầng thấp nhất, chịu trách nhiệm truyền các tín hiệu điện, ánh sáng hoặc sóng vô tuyến qua môi trường truyền dẫn vật lý như cáp đồng, cáp quang hoặc sóng không dây. Tầng này định nghĩa các đặc tính cơ học, điện học và chức năng của thiết bị như chuẩn đầu nối, điện áp tín hiệu, tốc độ truyền...

6.3.3. Mô hình TCP/IP

TCP/IP cũng là tên của một mô hình mạng, là bản mô hình hóa của tập hợp các giao thức (bộ giao thức TCP/IP), đóng vai trò nền tảng cho hoạt động của Internet ngày nay. Mô hình này được đặt tên theo chính bộ giao thức mà nó mô hình hóa.

Thiết kế của mô hình TCP/IP cũng dựa theo kiến trúc phân tầng. Tên gọi các tầng theo mô hình TCP/IP theo thứ tự từ thấp lên cao lần lượt là: tầng liên kết, tầng liên mạng, tầng giao vận, và tầng ứng dụng.

Tầng ứng dụng (Application layer) chịu trách nhiệm làm việc với dữ liệu người dùng. Các giao thức thuộc tầng này sẽ sử dụng một trong hai giao thức của tầng giao vận để truyền thông tin tới điểm cuối. Các giao thức tầng ứng dụng thường gặp: Telnet, FTP, DNS, SMTP, SNMP, TFTP, NFS, DHCP. Các giao thức tầng này được thực thi trực tiếp trong ứng dụng.

Tầng giao vận (Transport layer) chứa hai giao thức TCP (TA: Transmission Control Protocol, TCP) và UDP (TA: User Datagram Protocol, UDP), chịu trách nhiệm phục vụ tầng tầng ứng dụng và đảm bảo truyền thông điểm cuối - điểm cuối (ứng dụng với ứng dụng) giữa các máy trong mạng. Các giao thức này được thực thi ở dạng phần mềm và cài đặt trực tiếp trong hệ điều hành.

Tầng internet (Internet layer), còn được gọi là tầng mạng (Network layer), chịu trách nhiệm vận chuyển các gói tin qua liên mạng (các mạng kết nối với nhau qua thiết bị router). Tầng này chứa tất cả các giao thức định tuyến (IGMP, ICMP, RIP, OSPF) và giao thức vận chuyển dữ liệu người dùng IP. Các thiết bị hoạt động ở tầng này (router) có nhiệm vụ nhận gói tin IP, xác định điểm đến của gói tin IP, và chuyển gói tin IP về phía máy đích.

Tầng này cũng chứa các giao thức gửi và nhận thông báo lỗi và các thông điệp điều khiển. Các giao thức ở tầng này có thể được thực hiện ở cả phần cứng và phần mềm. Giao thức IP trên máy tính được thực hiện ở dạng hàm hệ thống, trên router được thực hiện ở phần cứng.

Thiết bị chủ yếu hoạt động ở tầng internet là bộ định tuyến router.

Tầng liên kết (Link layer), còn được gọi là tầng giao diện mạng (Network Interface layer), chịu trách nhiệm hoạt động với trình điều khiển thiết bị và các giao diện phần cứng để kết nối máy tính với môi trường truyền.

Một số giao thức tầng liên kết: ATM, Ethernet, PPP, Frame Relay, Token Ring, FDDI. Các giao thức ở tầng này được thực thi hầu hết trong phần cứng.

Tất cả các thiết bị mạng như hub/repeater, bridge/switch hoạt động ở tầng liên kết.

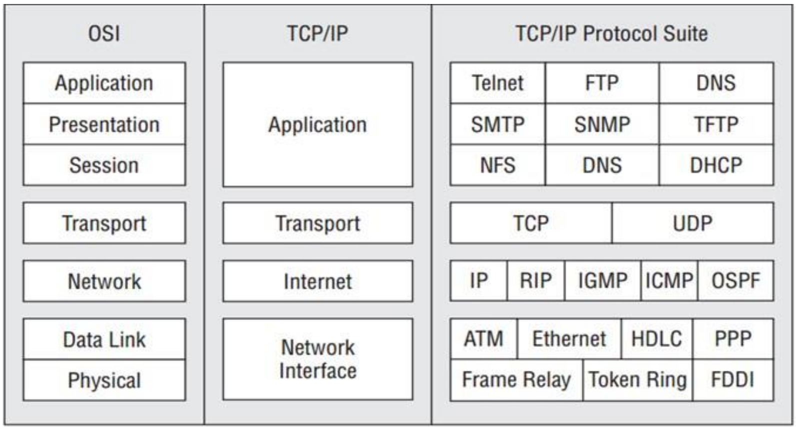
6.3.4. So sánh OSI với TCP/IP

Người ta thường so sánh mô hình OSI và TCP/IP. Thực tế là có sự tương đồng giữa các tầng của OSI và TCP/IP cả về vai trò nhiệm vụ cũng như tên gọi. Bảng dưới đây tóm lược sự giống và khác nhau giữa hai mô hình mạng này.

Bảng 6.2. So sánh mô hình OSI và TCP/IP

Mô hình OSI	Mô hình TCP/IP	Chức năng tương ứng
Tầng 7: Ứng dụng (Application)	Tầng 4: Ứng dụng (Application)	Giao tiếp người dùng, giao thức ứng dụng như HTTP, SMTP
Tầng 6: Trình diễn (Presentation)	Gộp vào tầng 4 TCP/IP	Mã hóa, nén dữ liệu
Tầng 5: Phiên (Session)	Gộp vào tầng 4 TCP/IP	Thiết lập, duy trì và kết thúc phiên giao tiếp
Tầng 4: Giao vận (Transport)	Tầng 3: Giao vận (Transport)	Quản lý truyền dữ liệu đầu-cuối, TCP/UDP
Tầng 3: Mạng (Network)	Tầng 2: Internet	Định tuyến, địa chỉ IP, ICMP, IP
Tầng 2: Liên kết dữ liệu (Data Link)	Tầng 1: Liên kết (Link)	Điều khiển truy cập, đóng khung
Tầng 1: Vật lý (Physical)	Tầng 1: Liên kết (Link)	Giao tiếp vật lý, truyền tín hiệu

Cần chú ý rằng, OSI được sinh ra mới mục đích mô tả bất kì loại mạng nào. Trong khi đó, TCP/IP được tạo ra chỉ để mô tả cho bộ giao thức TCP/IP có sẵn. Tuy nhiên, hiện nay chỉ còn duy nhất bộ giao thức TCP/IP được sử dụng làm chuẩn công nghiệp trong mạng máy tính. Vì vậy, bộ giao thức này cũng được sử dụng làm chuẩn để so sánh hai mô hình mạng.


OSI
Application
Presentation
Session
Transport
Network
Data Link
Physical
TCP/IP
Application
Transport
Internet
Network Interface
TCP/IP Protocol Suite
Telnet
FTP
DNS
SMTP
SNMP
TFTP
NFS
DNS
DHCP
TCP
UDP
IP
RIP
IGMP
ICMP
OSPF
ATM
Ethernet
HDLC
PPP
Frame Relay
Token Ring
FDDI

Hình 6.7. So sánh mô hình OSI, mô hình TCP/IP và bộ giao thức TCP/IP

Có thể thấy rằng, 3 tầng trên của OSI có cùng vai trò với tầng ứng dụng của TCP/IP. Tầng Transport của hai mô hình là tương đương nhau. Tầng Network của OSI tương đương tầng Internet của TCP/IP. Hai tầng dưới cùng của OSI gần tương đương với tầng truy cập mạng của TCP/IP.

Mô hình TCP/IP cũng được xem là một mô hình tham chiếu, mặc dù nó khác với mô hình OSI ở cách xây dựng và mục đích ban đầu. Mặc dù TCP/IP ban đầu không được xây dựng với mục đích lý thuyết như OSI, nhưng theo thời gian, nó đã trở thành một mô hình thực tế được tham chiếu rộng rãi, đặc biệt là trong các hệ thống mạng hiện đại (như Internet). Cụ thể:

- Nó mô tả các lớp chức năng tương đương với các tầng trong OSI.
- Nó xác định các giao thức cụ thể tại mỗi tầng (ví dụ: IP ở tầng mạng, TCP/UDP ở tầng giao vận, HTTP/FTP ở tầng ứng dụng).
- Do đó, nó cũng đóng vai trò "mô hình tham chiếu" trong thực tế, giúp hướng dẫn thiết kế, phát triển và triển khai các hệ thống mạng dựa trên TCP/IP.

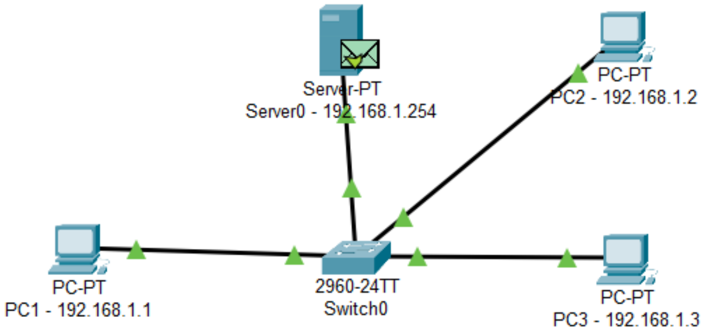
Như vậy có thể nói OSI là mô hình tham chiếu chuẩn hóa, mang tính lý thuyết, dùng để học, phân tích và thiết kế mạng, còn TCP/IP là mô hình tham chiếu thực tiễn, dùng trong việc xây dựng và vận hành mạng Internet thực tế.

Vì những lí do này, khi nghiên cứu về mạng máy tính, chúng ta phải tìm hiểu cả hai mô hình OSI và TCP/IP.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

Mở rộng mô hình mạng đã xây dựng ở bài thực hành 6.2 bằng cách bổ sung thêm một web server (192.168.1.254) như sau:



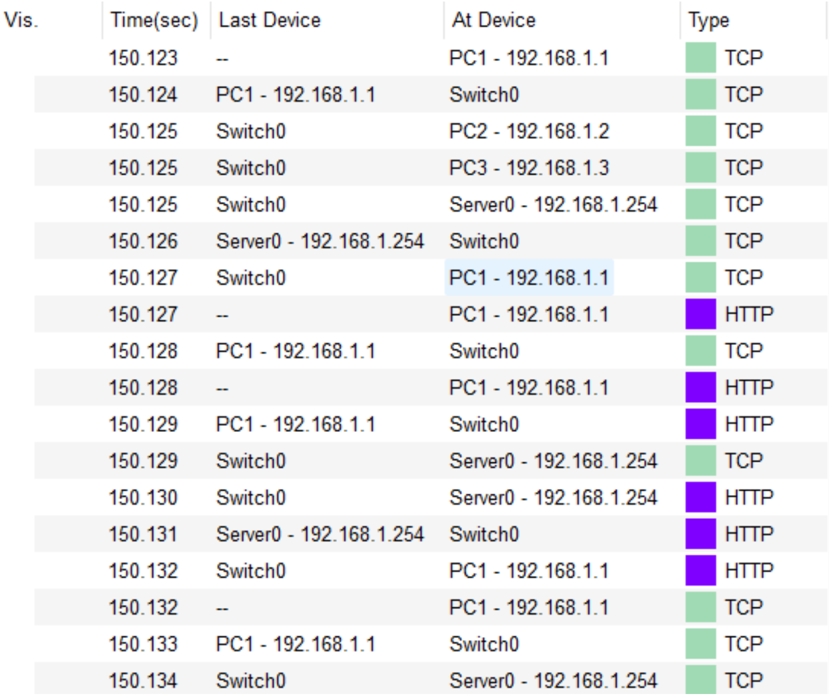
Bật lọc thêm cho các giao thức TCP và HTTP. Từ PC1 chạy web browser và truy cập vào web server trên Server0. Chạy mô phỏng để thu được kết quả.

Giải thích kết quả mô phỏng dựa trên kiến thức về toàn bộ quá trình đóng gói dữ liệu và truyền dữ liệu trong mạng đã học.

Câu hỏi lý thuyết

1) Tại sao trong cùng một mạng LAN, hai máy tính cần biết địa chỉ MAC của nhau để truyền dữ liệu, trong khi địa chỉ IP đã xác định được thiết bị đích? Giải thích vai trò của ARP trong trường hợp này.
2) Một máy tính có thể truyền dữ liệu tới một máy tính khác trong cùng mạng mà không cần địa chỉ IP không? Vì sao? Hãy lập luận dựa trên cơ chế hoạt động của ARP và tầng liên kết dữ liệu.

3) Giả sử hai máy tính A và B có địa chỉ IP ở các lớp khác nhau nhưng cùng kết nối với một switch. Liệu chúng có thể truyền thông tin cho nhau mà không cần router không? Vì sao?
4) Tại sao mỗi gói tin trong mạng chỉ có thể truyền qua một số router nhất định mà không thể đi tới mọi nơi trong Internet? Hãy giải thích vai trò của bảng định tuyến trong quá trình truyền thông liên mạng.
5) Giả sử một gói tin có địa chỉ IP đích là địa chỉ broadcast. Điều gì sẽ xảy ra khi router nhận được gói tin này? Vì sao router xử lý như vậy?
6) So sánh miền xung đột và miền quảng bá. Vì sao switch có thể giảm xung đột nhưng không chia nhỏ miền quảng bá, trong khi router làm được cả hai việc đó?
7) Một sinh viên cho rằng chế độ song công luôn tốt hơn bán song công và đơn công. Quan điểm này có đúng trong mọi trường hợp không? Hãy phân tích dựa trên các tình huống ứng dụng thực tế.
8) Khi cấu hình địa chỉ IP cho một thiết bị, nếu không khai báo Default Gateway thì điều gì sẽ xảy ra khi thiết bị này cố gắng gửi dữ liệu tới thiết bị ở mạng khác? Hãy giải thích quá trình này.
9) Tại sao trong thực tế người ta vẫn sử dụng các địa chỉ IP private mặc dù chúng không thể truy cập trực tiếp ra Internet? Nêu ít nhất hai lợi ích quan trọng.
10) Hãy giải thích tại sao mô hình OSI mặc dù không được dùng trực tiếp trong các hệ thống mạng thực tế nhưng vẫn giữ vai trò quan trọng trong giảng dạy và nghiên cứu về mạng máy tính.



Vis.	Time(sec)	Last Device	At Device	Type
	150.123	--	PC1 - 192.168.1.1	🟩 TCP
	150.124	PC1 - 192.168.1.1	Switch0	🟩 TCP
	150.125	Switch0	PC2 - 192.168.1.2	🟩 TCP
	150.125	Switch0	PC3 - 192.168.1.3	🟩 TCP
	150.125	Switch0	Server0 - 192.168.1.254	🟩 TCP
	150.126	Server0 - 192.168.1.254	Switch0	🟩 TCP
	150.127	Switch0	PC1 - 192.168.1.1	🟩 TCP
	150.127	--	PC1 - 192.168.1.1	🟪 HTTP
	150.128	PC1 - 192.168.1.1	Switch0	🟩 TCP
	150.128	--	PC1 - 192.168.1.1	🟪 HTTP
	150.129	PC1 - 192.168.1.1	Switch0	🟪 HTTP
	150.129	Switch0	Server0 - 192.168.1.254	🟩 TCP
	150.130	Switch0	Server0 - 192.168.1.254	🟪 HTTP
	150.131	Server0 - 192.168.1.254	Switch0	🟪 HTTP
	150.132	Switch0	PC1 - 192.168.1.1	🟪 HTTP
	150.132	--	PC1 - 192.168.1.1	🟩 TCP
	150.133	PC1 - 192.168.1.1	Switch0	🟩 TCP
	150.134	Switch0	Server0 - 192.168.1.254	🟩 TCP

BÀI 7. ỨNG DỤNG VÀ DỊCH VỤ MẠNG

Mục tiêu của bài học

Bài học giúp sinh viên hiểu và thực hành triển khai các dịch vụ mạng phổ biến như Web, DNS và FTP. Sinh viên sẽ cài đặt, cấu hình và kiểm thử các dịch vụ, đồng thời phân tích cấu trúc hệ thống và cơ chế hoạt động theo mô hình client-server.

Về kiến thức:

- Trình bày được nguyên lý hoạt động của HTTP, DNS và FTP.
- Mô tả được cấu trúc và cơ chế trao đổi giữa client và server.
- Phân biệt các khái niệm: tên miền, phân giải tên, kết nối điều khiển/dữ liệu.

Về kỹ năng:

- Cài đặt và cấu hình dịch vụ Web, DNS, FTP trên máy thật hoặc mô phỏng.
- Sử dụng FileZilla và Packet Tracer để kiểm thử và phân tích hoạt động dịch vụ mạng.

7.1. Thực hành - Cài đặt web server

Hệ thống web hiện nay có thể xem như một trong những loại phần mềm mạng phổ biến nhất. Trong một hệ thống web, giao thức HTTP đóng vai trò quan trọng hàng đầu, giúp các thành phần của hệ thống tương tác với nhau.

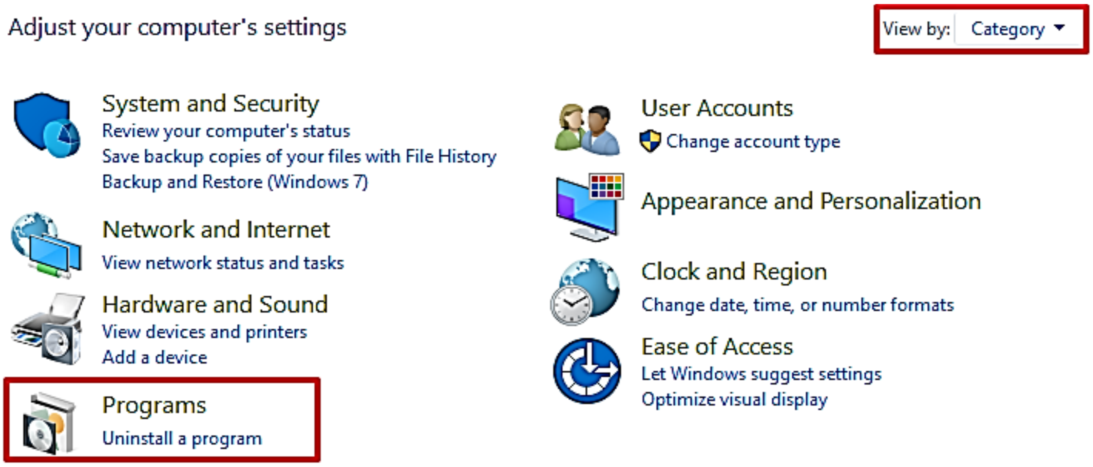
Trong bài học này chúng ta sẽ tìm hiểu về cách cài đặt và cấu hình cơ bản cho chương trình web server IIS và Apache. Để chuẩn bị cho bài học này, bạn cần chuẩn bị các phần mềm cần thiết sau:

- Trình duyệt: có thể sử dụng bất kỳ trình duyệt hiện đại nào, như Google Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari, v.v..
- Chương trình chủ: có thể sử dụng chương trình IIS (Internet Information Service) sẵn có trên Windows, hoặc cài đặt phần mềm mã mở Apache. Sau đây là hướng dẫn chi tiết để cài đặt hai loại chương trình chủ này.

7.1.1. Cài đặt IIS

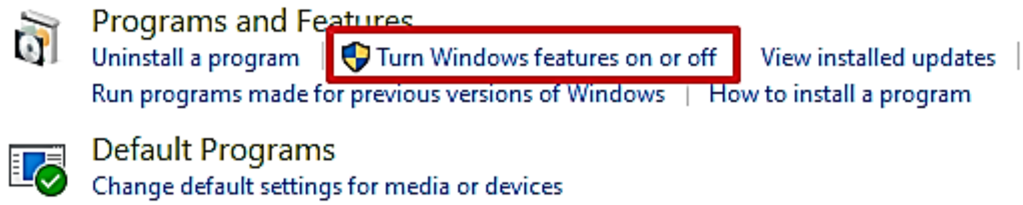
IIS là phần mềm máy chủ web dành riêng cho hệ điều hành Windows. Mặc định Windows không cài đặt sẵn IIS server. Bạn cần thực hiện một số thao tác để cài đặt phần mềm này:

(1) Mở Control Panel, chọn View by Category, và chọn Programs.


Adjust your computer's settings
View by: Category
System and Security
Review your computer's status
Save backup copies of your files with File History
Backup and Restore (Windows 7)
Network and Internet
View network status and tasks
Hardware and Sound
View devices and printers
Add a device
Programs
Uninstall a program
User Accounts
Change account type
Appearance and Personalization
Clock and Region
Change date, time, or number formats
Ease of Access
Let Windows suggest settings
Optimize visual display

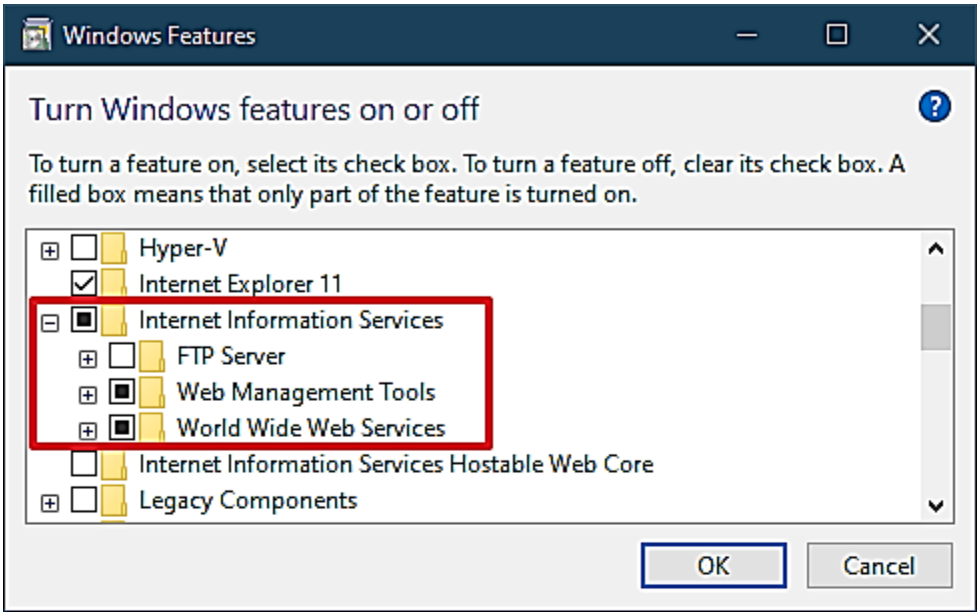
Hình 7.1. Lựa chọn Programs trong Control Panel

(2) Chọn Turn Windows features on or off


Programs and Features
Uninstall a program
Turn Windows features on or off
View installed updates
Run programs made for previous versions of Windows
How to install a program
Default Programs
Change default settings for media or devices

Hình 7.2. Lựa chọn Turn Windows features on or off

(3) Tìm trong danh sách và chọn Internet Information Services. Ấn OK để bắt đầu quá trình cài đặt.


Windows Features
Turn Windows features on or off

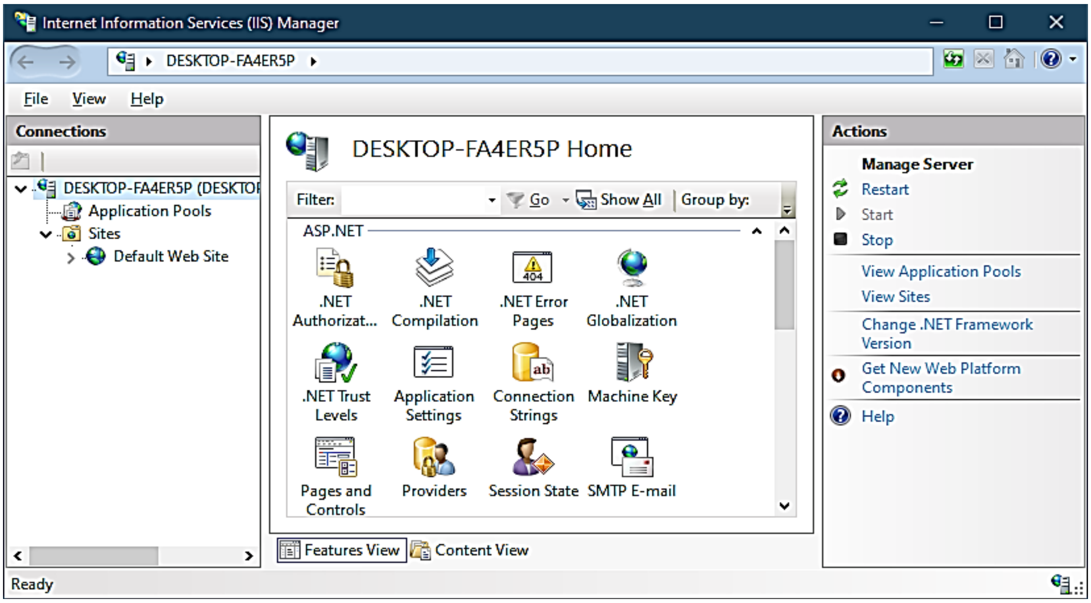
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ☐ Hyper-V
- ☑ Internet Explorer 11
- ■ Internet Information Services
  - ☐ FTP Server
  - ■ Web Management Tools
  - ■ World Wide Web Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components

OK
Cancel

Hình 7.3. Internet Information Services với các lựa chọn mặc định

(4) Để kiểm tra, tìm chạy phần mềm Internet Information Services (IIS) Manager bằng cách gõ IIS từ cửa sổ tìm kiếm của windows (ấn phím Windows, sau đó gõ IIS và ô tìm kiếm).


Internet Information Services (IIS) Manager
DESKTOP-FA4ER5P
File
View
Help
Connections
DESKTOP-FA4ER5P (DESKTOP
Application Pools
Sites
Default Web Site
DESKTOP-FA4ER5P Home
Filter:
Go
Show All
Group by:
ASP.NET
.NET Authorizat...
.NET Compilation
.NET Error Pages
.NET Globalization
.NET Trust Levels
Application Settings
Connection Strings
Machine Key
Pages and Controls
Providers
Session State
SMTP E-mail
Features View
Content View
Actions
Manage Server
Restart
Start
Stop
View Application Pools
View Sites
Change .NET Framework Version
Get New Web Platform Components
Help
Ready

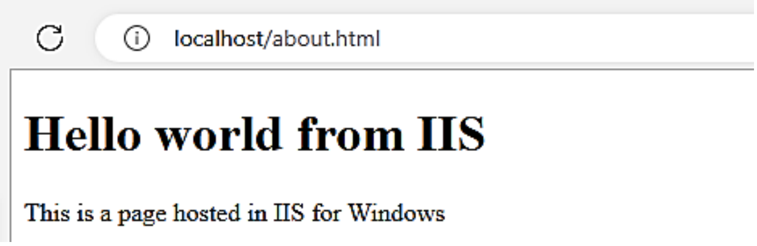
Hình 7.4. Giao diện IIS Manager

Để thử nghiệm server mới trên máy cục bộ, bạn có thể mở trình duyệt và nhập địa chỉ http://localhost hoặc http://127.0.0.1.

(5) Để tạo trang web trong IIS, bạn mở thư mục C:\inetpub\wwwroot và tạo mới file about.html. Nhập nội dung như sau cho file này:

```
<!DOCTYPE html>
<html>
<body>
<h1>Hello world from IIS</h1>
<p>This is a page hosted in IIS for Windows</p>
</body>
</html>
```

Mở trình duyệt và truy xuất trang qua địa chỉ `http://localhost/about.html`, bạn sẽ thu được kết quả như Hình 7.5.


localhost/about.html

Hello world from IIS

This is a page hosted in IIS for Windows

Hình 7.5. Mở trang web tự tạo trong trình duyệt

Mặc định các file của website IIS được lưu trong thư mục C:\inetpub\wwwroot. Bạn có thể đặt các file html vào thư mục này và truy xuất từ trình duyệt.

7.1.2. Cài đặt Apache server

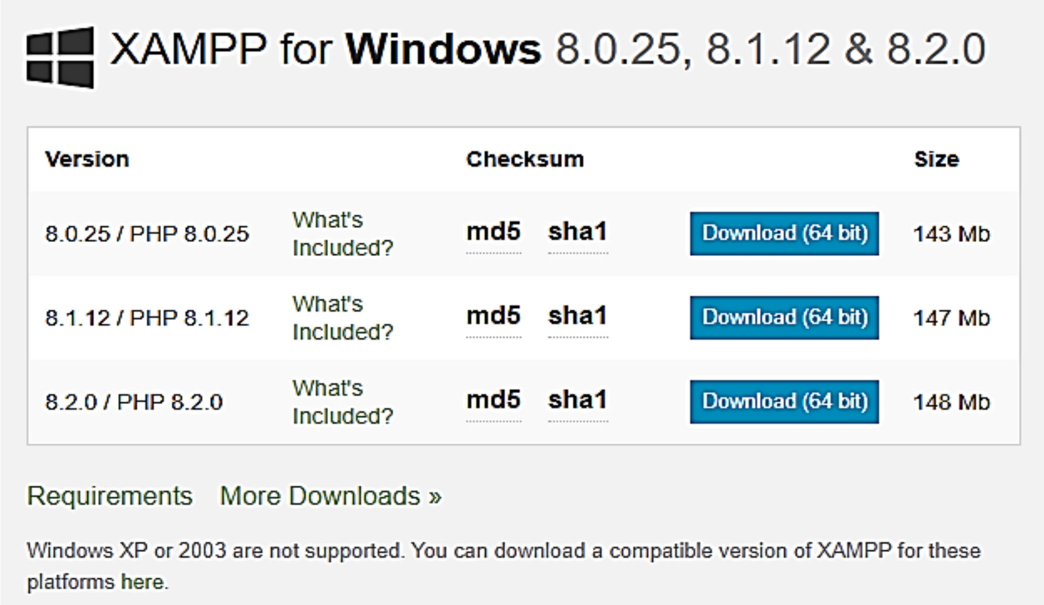
Apache là một phần mềm máy chủ web mã mở đặc biệt phổ biến trên hệ điều hành Linux. Tuy nhiên, việc cài đặt Apache riêng rẽ tương đối phức tạp vì phải thực hiện nhiều thao tác cấu hình. Đặc biệt, Apache thường được sử dụng chung với các phần mềm máy chủ khác như MySQL / MariaDb. Do vây cộng đồng mã mở tạo ra các bộ phần mềm máy chủ để đơn giản hóa việc cài đặt chúng. Các bộ phần mềm mã mở thường gặp là XAMPP, AMPP, LAMPP.

XAMPP là bộ chương trình chủ mã mở, bao gồm chương trình chủ web Apache, server cơ sở dữ liệu MySQL (MariaDb), trình thông dịch lệnh PHP và Perl. Bộ phần mềm này có thể cài đặt trên tất cả các hệ điều thành PC hiện nay (Linux, Windows, MacOs).

Để cài đặt XAMPP, bạn thực hiện theo các bước sau:

(1) Tải bộ cài từ địa chỉ https://www.apachefriends.org/download.html

Tùy vào hệ điều hành đang sử dụng, bạn chọn phiên bản tương ứng.



XAMPP for **Windows** 8.0.25, 8.1.12 & 8.2.0

Version		Checksum			Size
8.0.25 / PHP 8.0.25	What's Included?	md5	sha1	Download (64 bit)	143 Mb
8.1.12 / PHP 8.1.12	What's Included?	md5	sha1	Download (64 bit)	147 Mb
8.2.0 / PHP 8.2.0	What's Included?	md5	sha1	Download (64 bit)	148 Mb

Requirements More Downloads »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

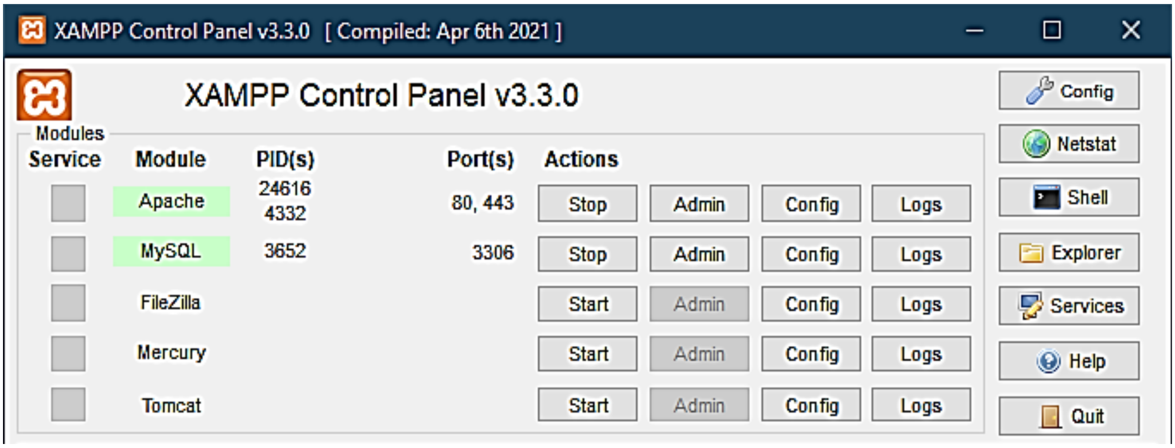
Hình 7.6. Lựa chọn phiên bản XAMPP trên website

(2) Trên windows, bạn cài đặt nó như một phần mềm bình thường. Giả sử bạn cài phần mềm này vào thư mục E:\XAMPP.

(3) Sau khi cài đặt xong, bạn vào thư mục này và chạy chương trình xampp-control.exe. Giao diện chương trình này xem ở Hình 7.7.

(4) Bạn chạy Apache server bằng cách click vào nút Start tương ứng.

(5) Để tạo trang web mới trong XAMPP - Apache, bạn tạo file html trong thư mục XAMPP\htdocs.

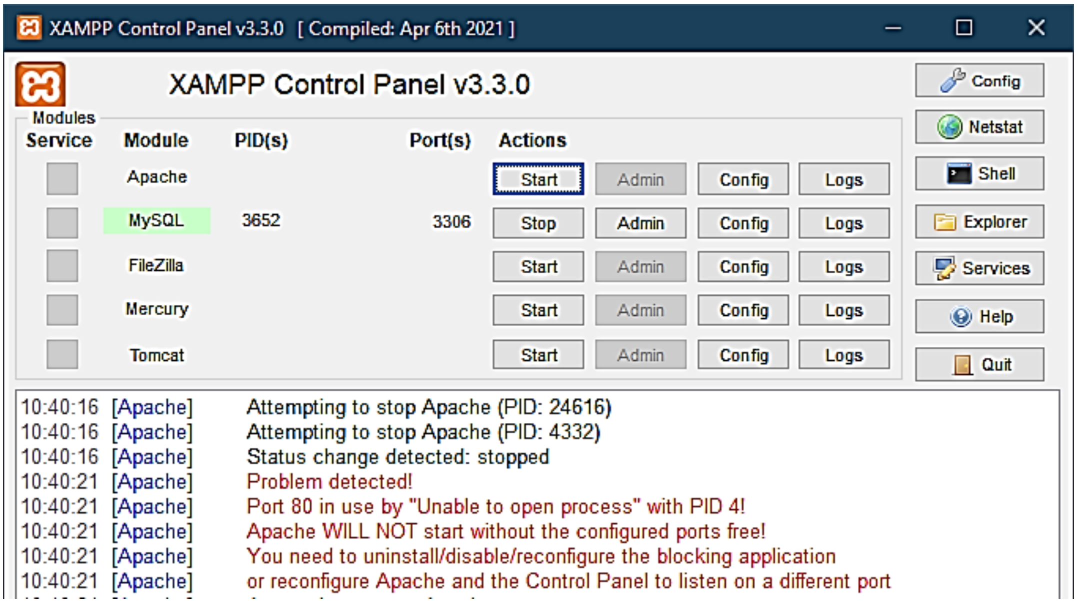

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
24616
4332
80, 443
MySQL
3652
3306
FileZilla
Mercury
Tomcat
Stop
Start
Admin
Config
Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit

Hình 7.7. Giao diện quản lý XAMPP Control Panel

Mặc định các website của Apache được lưu trong thư mục XAMPP\htdocs. Vai trò của thư mục này cũng tương tự như C:\inetpub\wwwroot của IIS. Bạn có thể đặt các file html vào thư mục htdocs và truy xuất nó qua trình duyệt.

7.1.3. Lưu ý khi chạy server

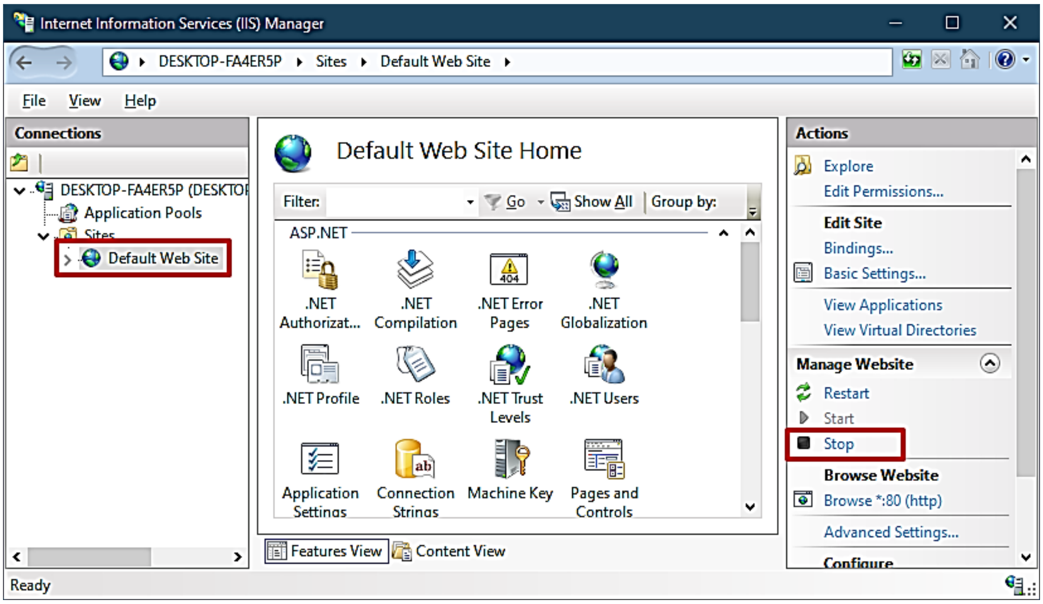
Lưu ý, do các chương trình chủ web mặc định sử dụng cổng TCP/80, bạn không thể đồng thời chạy cả hai chương trình chủ web. Nếu IIS đang hoạt động, khi khởi động Apache sẽ báo lỗi như Hình 7.8.


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
MySQL
3652
3306
FileZilla
Mercury
Tomcat
Start
Stop
Admin
Config
Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit

```
10:40:16  [Apache]     Attempting to stop Apache (PID: 24616)
10:40:16  [Apache]     Attempting to stop Apache (PID: 4332)
10:40:16  [Apache]     Status change detected: stopped
10:40:21  [Apache]     Problem detected!
10:40:21  [Apache]     Port 80 in use by "Unable to open process" with PID 4!
10:40:21  [Apache]     Apache WILL NOT start without the configured ports free!
10:40:21  [Apache]     You need to uninstall/disable/reconfigure the blocking application
10:40:21  [Apache]     or reconfigure Apache and the Control Panel to listen on a different port
```

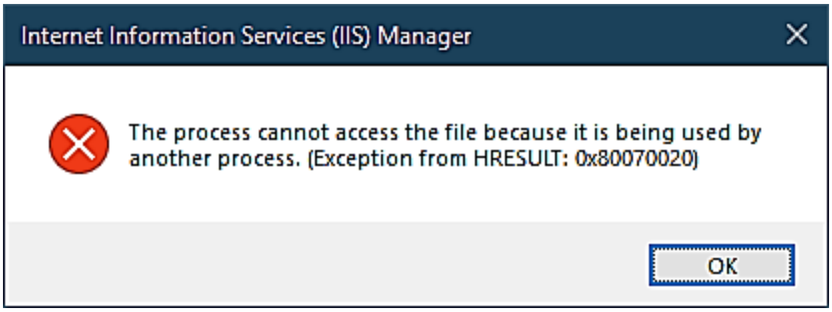
Hình 7.8. Lỗi khởi động Apache khi có một web server khác đang hoạt động

Khi này, bạn cần tắt IIS trước khi chạy Apache như Hình 7.9.


Internet Information Services (IIS) Manager
DESKTOP-FA4ER5P ▸ Sites ▸ Default Web Site ▸
File View Help
Connections
DESKTOP-FA4ER5P (DESKTOP
Application Pools
Sites
Default Web Site
Default Web Site Home
Filter:
Go
Show All
Group by:
ASP.NET
.NET Authorizat...
.NET Compilation
.NET Error Pages
.NET Globalization
.NET Profile
.NET Roles
.NET Trust Levels
.NET Users
Application Settings
Connection Strings
Machine Key
Pages and Controls
Features View
Content View
Actions
Explore
Edit Permissions...
Edit Site
Bindings...
Basic Settings...
View Applications
View Virtual Directories
Manage Website
Restart
Start
Stop
Browse Website
Browse *:80 (http)
Advanced Settings...
Configure
Ready

Hình 7.9. Tắt IIS trong giao diện quản lý

Tương tự, nếu Apache đang hoạt động, bật IIS cũng sẽ báo lỗi:


Internet Information Services (IIS) Manager
OK

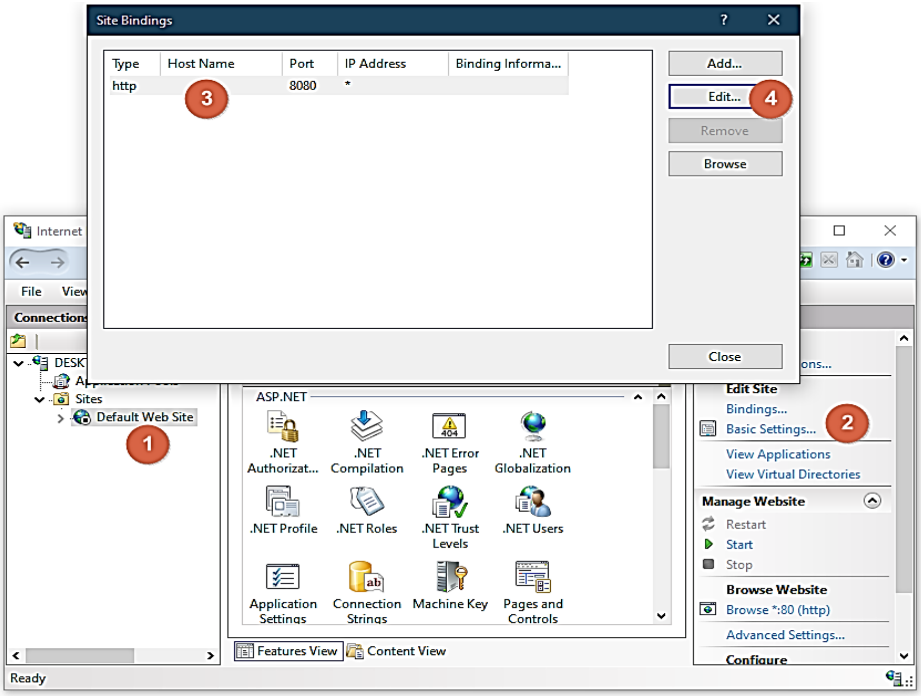
The process cannot access the file because it is being used by another process. (Exception from HRESULT: 0x80070020)

Hình 7.10. Báo lỗi trên IIS khi có một web server khác đang hoạt động

Khi này cần tắt Apache rồi mới khởi động IIS.

Một cách khác cho phép hai chương trình server này chạy đồng thời là thay đổi cổng TCP của một trong hai chương trình.

Đối với IIS bạn thực hiện theo các bước như sau để đổi cổng mặc định về 8080 (Hình 7.11).

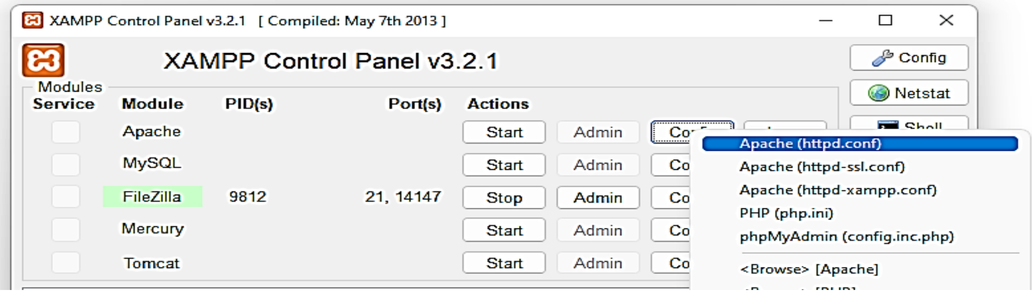

Site Bindings
Add...
Edit...
Remove
Browse
Close
3
4
Internet
File
View
Connections
DESK
Sites
Default Web Site
1
ASP.NET
.NET Authorizat...
.NET Compilation
.NET Error Pages
.NET Globalization
.NET Profile
.NET Roles
.NET Trust Levels
.NET Users
Application Settings
Connection Strings
Machine Key
Pages and Controls
Features View
Content View
Edit Site
Bindings...
Basic Settings...
2
View Applications
View Virtual Directories
Manage Website
Restart
Start
Stop
Browse Website
Browse *:80 (http)
Advanced Settings...
Configure
Ready

Type	Host Name	Port	IP Address	Binding Informa...
http		8080	*	

Hình 7.11. Thay đổi cổng mặc định trên IIS

Đối với Apache, bạn thay đổi bằng cách sau:

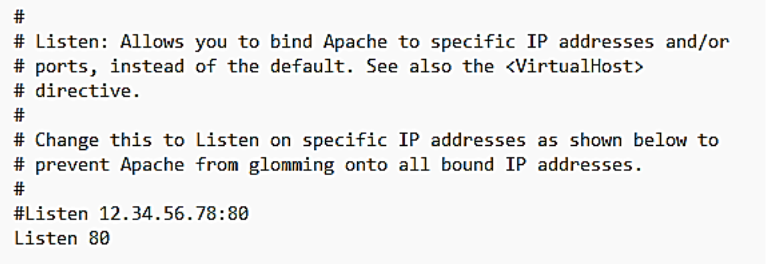
Bấm nút Config tương ứng của Apache và chọn Apache (httpd.conf).


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Config
Netstat
Shell
Modules
Apache (httpd.conf)
Apache (httpd-ssl.conf)
Apache (httpd-xampp.conf)
PHP (php.ini)
phpMyAdmin (config.inc.php)
<Browse> [Apache]

Service	Module	PID(s)	Port(s)	Actions		
	Apache			Start	Admin	Co
	MySQL			Start	Admin	Co
	FileZilla	9812	21, 14147	Stop	Admin	Co
	Mercury			Start	Admin	Co
	Tomcat			Start	Admin	Co

Hình 7.12. Thay đổi cấu hình Apache

Cửa sổ editor sẽ mở ra cùng với file cấu hình của Apache. Bạn tìm đến dòng có nội dung như sau:



```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

Hình 7.13. Dòng cấu hình cổng mặc định trong file cấu hình Apache

Thay thế giá trị 80 bằng một số cổng TCP khác.

Số cổng mới sẽ được sử dụng khi bạn khởi động lại Apache.

7.2. Dịch vụ DNS

7.2.1. Tên miền và hệ thống tên miền

a) Khái niệm

Tên miền (Domain Name) là một chuỗi ký tự có cấu trúc, dùng để định danh và xác định một địa chỉ duy nhất trên Internet. Tên miền thường được sử dụng để gọi tên và truy cập các tài nguyên mạng như website, dịch vụ email, máy chủ ứng dụng v.v.

Hệ thống tên miền - DNS (Domain Name System) là một hệ thống phân tán toàn cầu, cung cấp cơ chế *phân giải tên miền* sang địa chỉ IP tương ứng và ngược lại. Nhờ có DNS, người dùng có thể truy cập các tài nguyên mạng thông qua tên dễ nhớ thay vì phải ghi nhớ các địa chỉ IP dạng số.

Quá trình **phân giải tên** (name resolution) được thực hiện bởi các chương trình và dịch vụ DNS trên máy chủ DNS (DNS Server), đáp ứng truy vấn từ máy khách (client) thông qua trình phân giải tên (Resolver).

Lợi ích của hệ thống DNS:

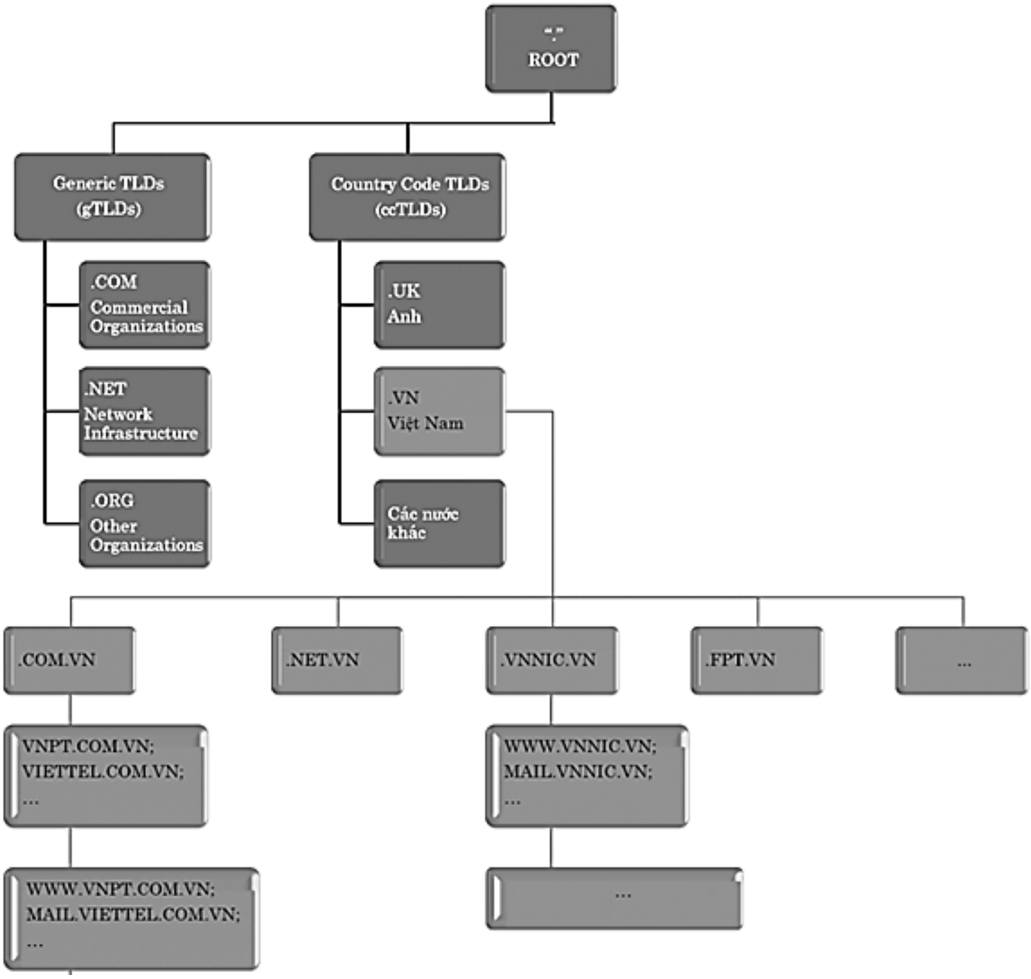
- Cho phép các tài nguyên mạng dễ dàng định danh và truy cập mà không phụ thuộc địa chỉ IP.
- Hỗ trợ ánh xạ một tên miền tới nhiều địa chỉ IP (load balancing).
- Cho phép nhiều tên miền cùng trỏ về một địa chỉ IP (aliasing).
- Hỗ trợ một máy chủ có thể đảm nhận nhiều vai trò hoặc phục vụ nhiều dịch vụ.
- Cho phép quản lý linh hoạt tài nguyên mạng trong môi trường phân tán.

b) Cấu trúc tên miền

Một tên miền được tạo thành từ các **nhãn** (label) không rỗng phân cách nhau bằng dấu chấm (.). Mỗi nhãn có thể chứa các chữ cái ASCII từ a đến z (không phân biệt hoa thường), chữ số từ 0 đến 9, và dấu gạch ngang (-). Dấu gạch ngang không được xuất hiện ở đầu hoặc cuối của nhãn. Tổng chiều dài của một tên miền không được vượt quá 253 ký tự. Tổng số nhãn tối đa là 127. Chiều dài của nhãn NÊN trong khoảng từ 1-63 ký tự.

Tên miền có cấu trúc phân cấp, giảm dần từ phải qua trái:

- Nhãn ngoài cùng bên phải là **tên miền cấp cao** (Top-Level Domain - TLD), ví dụ: .com, .org, .vn.
- Dưới TLD là **tên miền cấp hai** (Second-Level Domain - SLD), ví dụ: example.com.
- Tiếp theo là các **miền con** (subdomain), ví dụ: www.example.com.


"."
ROOT
Generic TLDs (gTLDs)
Country Code TLDs (ccTLDs)
.COM Commercial Organizations
.NET Network Infrastructure
.ORG Other Organizations
.UK Anh
.VN Việt Nam
Các nước khác
.COM.VN
.NET.VN
.VNNIC.VN
.FPT.VN
...
VNPT.COM.VN; VIETTEL.COM.VN; ...
WWW.VNNIC.VN; MAIL.VNNIC.VN; ...
WWW.VNPT.COM.VN; MAIL.VIETTEL.COM.VN; ...
...

Hình 7.14. Cấu trúc phân cấp của hệ thống tên miền

Tên miền cấp cao bao gồm:

- gTLD (generic TLD): dùng chung như .com, .net, .org, v.v.
- ccTLD (country code TLD): mã quốc gia theo chuẩn ISO-3166 như .vn, .jp, .uk.

Tên miền gốc (root domain) được biểu diễn bằng dấu chấm (.) và đứng ở đỉnh cây phân cấp. Toàn bộ hệ thống tên miền tạo thành một cấu trúc cây logic với khả năng phân chia không giới hạn số tầng (tối đa 127).

Ví dụ:

- `www.example.com` là một miền con của `example.com`, nằm dưới TLD .com.
- `edu.vn` là miền cấp hai thuộc ccTLD `.vn`, dành cho các tổ chức giáo dục tại Việt Nam.

c) Tên miền quốc gia Việt Nam

Theo Thông tư số 09/2008/TT-BTTTT, hệ thống tên miền quốc gia Việt Nam được phân cấp như sau:

- Tên miền cấp cao nhất: .vn
- Tên miền cấp hai: gồm các tên miền dùng chung theo lĩnh vực (như com.vn, gov.vn, edu.vn, net.vn, org.vn, v.v.) hoặc theo địa giới hành chính (như hanoi.vn, danang.vn).

Mỗi tên miền có quy định sử dụng riêng, ví dụ:

- com.vn: dành cho tổ chức/cá nhân hoạt động thương mại.
- edu.vn: dành cho giáo dục, đào tạo.
- gov.vn: dành cho cơ quan Nhà nước.
- name.vn: dành cho cá nhân.

d) Đăng ký tên miền

Người dùng Internet có thể đăng ký tên miền thông qua các *nhà đăng ký tên miền* (Domain Name Registrars). Việc đăng ký thường đi kèm chi phí và có thời hạn duy trì. Chủ thể đăng ký tên miền có quyền sử dụng độc quyền trong thời gian đăng ký, nhưng không sở hữu tuyệt đối tên đó.

Tên miền khi mới đăng ký chưa gắn với địa chỉ IP. Để sử dụng, cần khai báo ánh xạ từ tên miền tới IP thông qua hệ thống máy chủ DNS. Khi tên miền được ánh xạ với ít nhất một địa chỉ IP, nó trở thành một tên máy (hostname).

7.2.2. Cơ chế hoạt động của DNS

Hệ thống DNS hoạt động theo mô hình Client-Server bao gồm hai thành phần:

- Resolver (Client): là thành phần trên máy người dùng, chịu trách nhiệm khởi tạo truy vấn để phân giải tên miền sang địa chỉ IP. Resolver là các hàm thư viện dùng để tạo các truy vấn (query) và gửi chúng qua đến Name Server sử dụng giao thức DNS
- Name Server (Server): là máy chủ chứa thông tin bản ghi tên miền, có thể phản hồi truy vấn từ Resolver. Name Server cũng chia thành cấp độ tương tự như tên miền, bao gồm: Root DNS Server (chứa thông tin về các TLD Server), TLD DNS Server (chứa thông tin về các Authoritative Name Server), Authoritative Name Server (chứa dữ liệu thực sự về tên miền và địa chỉ IP).

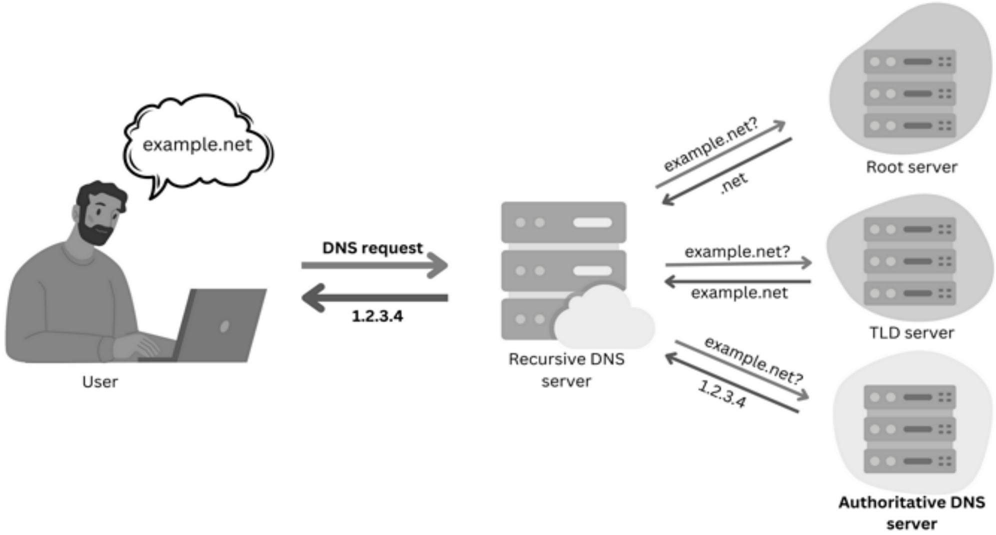
Quá trình phân giải tên miền diễn ra như sau:

1) Trình duyệt hoặc ứng dụng gửi yêu cầu phân giải tên miền đến Resolver cục bộ (thường do hệ điều hành cung cấp).
2) Resolver kiểm tra cache (bộ nhớ đệm) để xem tên miền đã được phân giải trước đó chưa. Nếu có, kết quả được trả về ngay.
3) Nếu không tìm thấy trong cache, Resolver gửi truy vấn đến máy chủ DNS đệ quy (recursive DNS server).
4) Máy chủ này lần lượt truy vấn tới các cấp máy chủ:

- Root DNS Server: yêu cầu cung cấp thông tin về TLD Server phù hợp.

- TLD DNS Server: yêu cầu cung cấp thông tin về Authoritative Name Server.
- Authoritative Name Server: yêu cầu cung cấp địa chỉ IP chính xác của tên miền.

5) Kết quả địa chỉ IP được trả về theo chiều ngược lại cho Resolver và sau đó cho ứng dụng.



Hình 7.15. Quy trình truy vấn địa chỉ IP từ tên miền

Có hai kiểu truy vấn DNS chính:

- **Truy vấn đệ quy** (Recursive Query): Resolver yêu cầu máy chủ DNS phải trả về kết quả cuối cùng (địa chỉ IP) hoặc thông báo lỗi.
- **Truy vấn lặp** (Iterative Query): Resolver sẽ tự xử lý từng bước, truy vấn tuần tự từ Root, TLD đến các máy chủ DNS có thẩm quyền.

Hệ thống DNS cung cấp hai cơ chế ánh xạ tên miền với địa chỉ IP, khác nhau ở tính chất cập nhật địa chỉ IP:

- DNS tĩnh (Static DNS): ánh xạ tên miền cố định với IP, phù hợp với hệ thống ổn định, không đổi IP thường xuyên. Việc cấu hình DNS tĩnh thường được thực hiện trên giao diện quản trị của nhà cung cấp dịch vụ hosting.
- DNS động (Dynamic DNS, DDNS): hỗ trợ cập nhật địa chỉ IP động cho các máy chủ có IP thay đổi theo thời gian, phù hợp với người dùng cá nhân hoặc doanh nghiệp nhỏ dùng kết nối Internet không cố định. Cơ chế này giúp duy trì kết nối ổn định và thuận tiện ngay cả khi địa chỉ IP của thiết bị thay đổi thường xuyên.

7.3. Thực hành - Cài đặt và sử dụng dịch vụ truyền file

FTP (File Transfer Protocol) là tên một loại hệ thống phần mềm mạng, đồng thời cũng là tên giao thức được sử dụng trong phần mềm này. Hệ thống phần mềm FTP cho phép người dùng truy cập và làm việc với hệ thống file trên một máy tính khác, tương tự như cách người dùng làm việc với hệ thống file trên máy tính cục bộ.

Hệ thống phần mềm FTP bao gồm hai phần mềm: phần mềm client cho người dùng, và phần mềm server chạy trên máy chủ. Hai phần mềm này tương tác với nhau thông qua giao thức FTP. Giao thức này sử dụng dịch vụ truyền thông của TCP.

FTP được sử dụng rất phổ biến bởi lập trình viên cũng như quản trị viên hệ thống. FTP giúp lập trình viên nhanh chóng truy xuất file của phần mềm trên server từ máy dev. Nó cũng giúp quản trị viên lưu dự phòng dữ liệu khối lượng lớn.

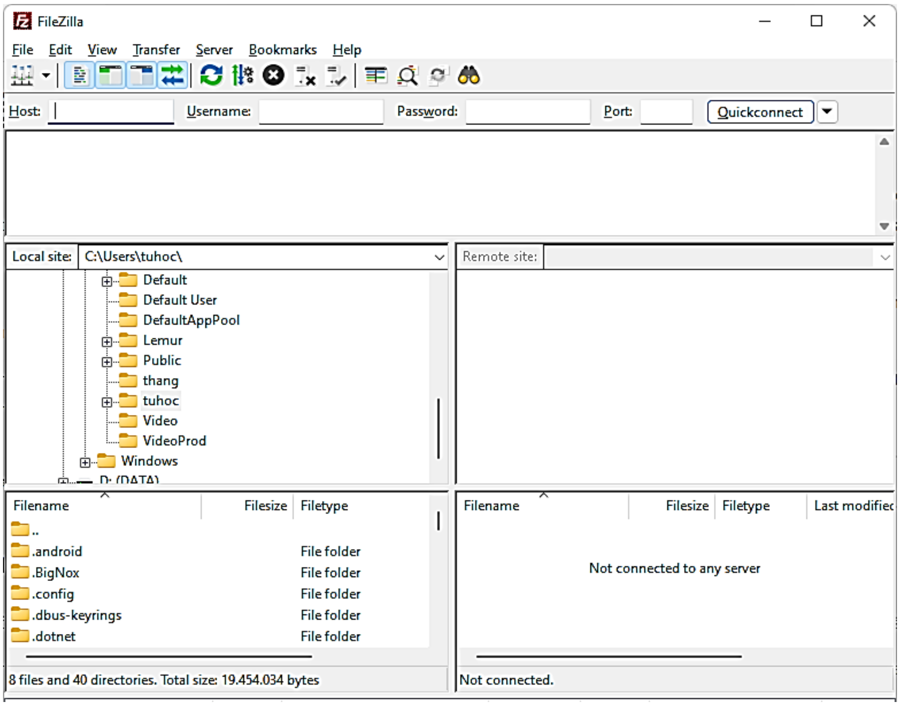
Hầu hết các nhà cung cấp dịch vụ hosting đều cung cấp luôn FTP server. Khi đó có thể sử dụng FileZilla client để làm việc với thư mục trên server đơn giản như làm việc với máy cục bộ.

Hiện nay có khá nhiều phần mềm FTP client, như FileZilla, FireFTP (Firefox add-on), WinSCP, Cyberduck, CoffeeCup free FTP. Tương tự cũng có rất nhiều phần mềm FTP server như FileZilla, Microsoft IIS, Cerberus, pyftpdlib (library), CrushFTP.

Trong bài học này chúng ta sẽ sử dụng bộ client và server của Filezilla.

7.3.1. Cài đặt FileZilla client và server

a) Cài đặt FileZilla client


FileZilla
File
Edit
View
Transfer
Server
Bookmarks
Help
Host:
Username:
Password:
Port:
Quickconnect
Local site:
Remote site:
Filename
Filesize
Filetype
Last modified

Local site: C:\Users\tuhoc\

Default
Default User
DefaultAppPool
Lemur
Public
thang
tuhoc
Video
VideoProd
Windows
D: (DATA)

..
.android File folder
.BigNox File folder
.config File folder
.dbus-keyrings File folder
.dotnet File folder

8 files and 40 directories. Total size: 19.454.034 bytes

Not connected to any server

Not connected.

Hình 7.16. Giao diện phần mềm FileZilla client

FileZilla client là phần mềm miễn phí được cung cấp tại địa chỉ:

https://filezilla-project.org/download.php

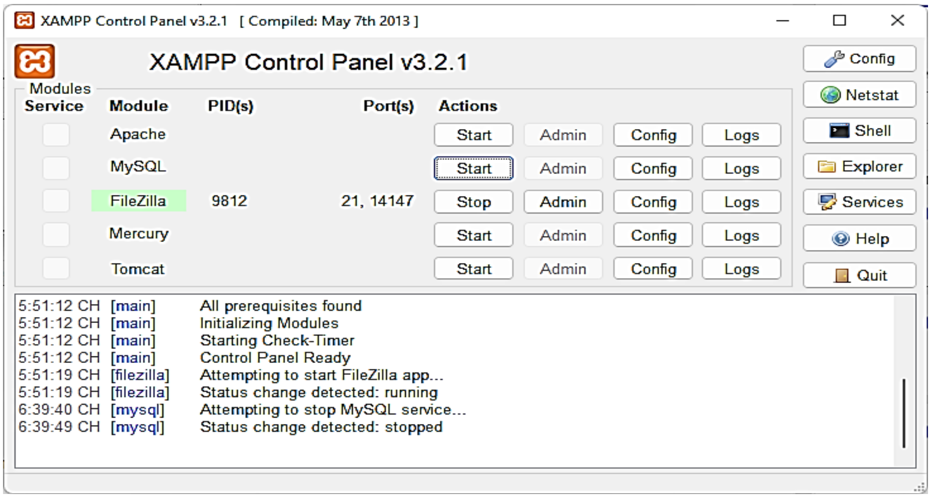
Tải phần mềm này về máy và cài đặt tương tự như các ứng dụng windows khác.

b) Cài đặt FileZilla server

FileZilla server cũng được cung cấp miễn phí. Bạn có thể download nó từ đường link sau: https://filezilla-project.org/download.php?type=server

Tuy nhiên, trong bài học trước bạn đã cài đặt bộ phần mềm server XAMPP. Trong bộ XAMPP đã cài đặt sẵn FileZilla server. Vì vậy bạn có thể sử dụng luôn mà không cần cài đặt server riêng.

Từ giao diện XAMPP Control Panel bạn có thể bật FileZilla server:


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
☐	Apache			Start	Admin	Config	Logs
☐	MySQL			Start	Admin	Config	Logs
☐	FileZilla	9812	21, 14147	Stop	Admin	Config	Logs
☐	Mercury			Start	Admin	Config	Logs
☐	Tomcat			Start	Admin	Config	Logs

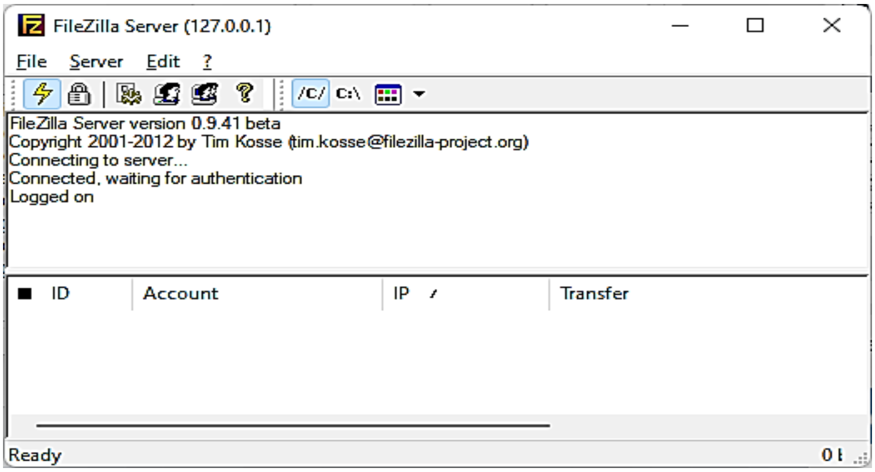
```
5:51:12 CH  [main]      All prerequisites found
5:51:12 CH  [main]      Initializing Modules
5:51:12 CH  [main]      Starting Check-Timer
5:51:12 CH  [main]      Control Panel Ready
5:51:19 CH  [filezilla] Attempting to start FileZilla app...
5:51:19 CH  [filezilla] Status change detected: running
6:39:40 CH  [mysql]     Attempting to stop MySQL service...
6:39:49 CH  [mysql]     Status change detected: stopped
```

Hình 7.17. Khởi động FileZilla server từ giao diện quả lý của XAMPP

7.3.2. Cấu hình FileZilla server

Để client có thể kết nối và làm việc với server, chúng ta cần tạo người dùng, phân quyền và cung cấp thư mục làm việc cho người dùng đó. Quy trình làm như sau:

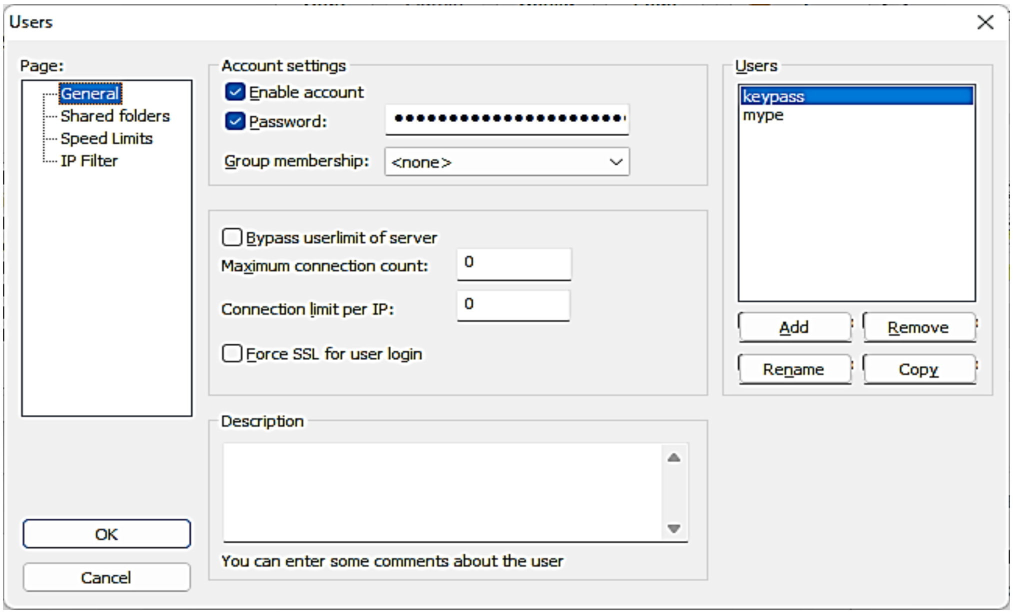
(1) Từ giao diện XAMPP Control Panel, click nut Admin tương ứng với FileZilla. Bạn sẽ thấy giao diện như sau:


FileZilla Server (127.0.0.1)
File
Server
Edit
?
/C/
C:\
ID
Account
IP
Transfer
Ready
0 b

```
FileZilla Server version 0.9.41 beta
Copyright 2001-2012 by Tim Kosse (tim.kosse@filezilla-project.org)
Connecting to server...
Connected, waiting for authentication
Logged on
```

Hình 7.18. Giao diện ứng dụng quản trị của FileZilla server.

(2) Từ giao diện này chọn menu Edit -> Users.



Users
Page:
General
Shared folders
Speed Limits
IP Filter
OK
Cancel
Account settings
Description
Users
Add
Remove
Rename
Copy

☑ Enable account
☑ Password: ••••••••••••••••••••••
Group membership: <none>

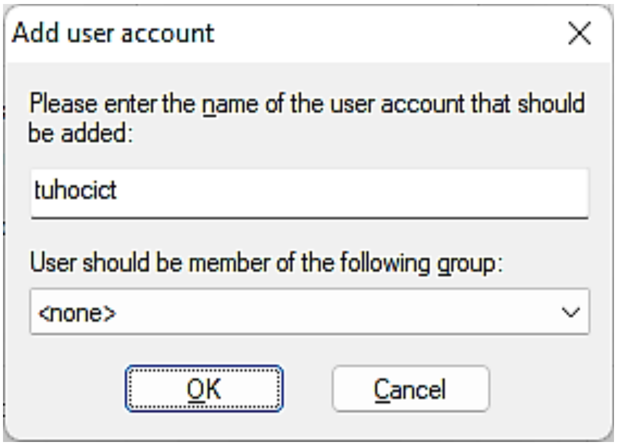
☐ Bypass userlimit of server
Maximum connection count: 0
Connection limit per IP: 0
☐ Force SSL for user login

You can enter some comments about the user

keypass
mype

Hình 7.19. Giao diện quản lý người dùng của FileZilla server

(3) Ấn nút "Add" trong group Users bên tay phải để thêm người dùng mới.



Add user account
OK
Cancel

Please enter the name of the user account that should be added:

tuhocict

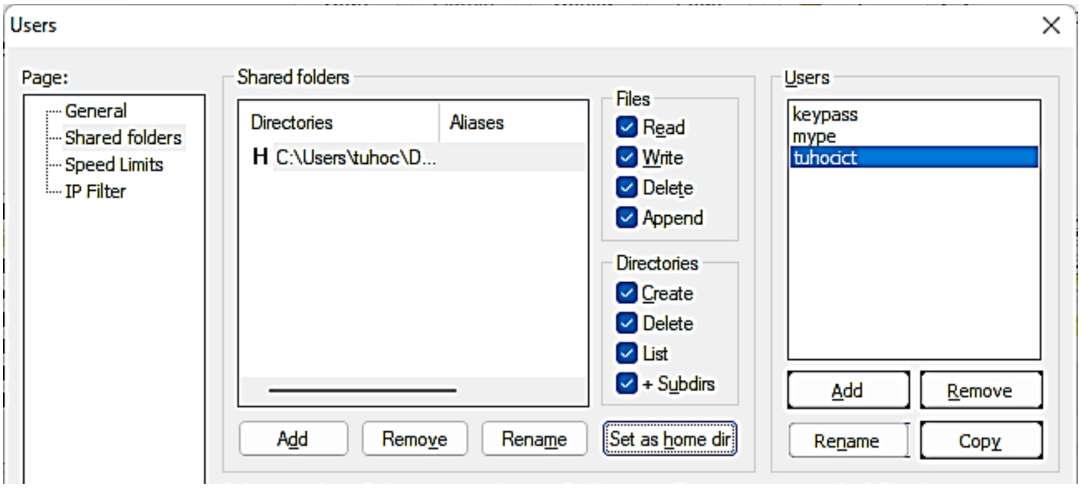
User should be member of the following group:

<none>

Hình 7.20. Tạo tài khoản FTP mới

(4) Sau khi đóng cửa sổ Add user account, ấn tiếp nút Add trong nhóm Shared folders để thêm thư mục làm việc cho user mới. Hãy chọn một thư mục bất kỳ trên ổ đĩa và ấn OK.

Khi quay trở lại cửa sổ Users, hãy click chọn các quyền đối với Files và Directories (Hình 13.6). Đối với file có các quyền Read (đọc file), Write (tạo file mới), Delete (xóa file), Append (cập nhật file). Đối với thư mục có thể chọn các quyền Create (tạo thư mục), Delete (xóa thư mục), List (lấy danh sách file), + Subdirs (tạo thư mục con). Sau bước này, người dùng mới đã có toàn quyền truy cập thư mục làm việc đã chọn.


Users
Page:
General
Shared folders
Speed Limits
IP Filter
Shared folders
Directories
Aliases
Files
Directories
Users
keypass
mype
tuhocict
Add
Remove
Rename
Set as home dir
Add
Remove
Rename
Copy

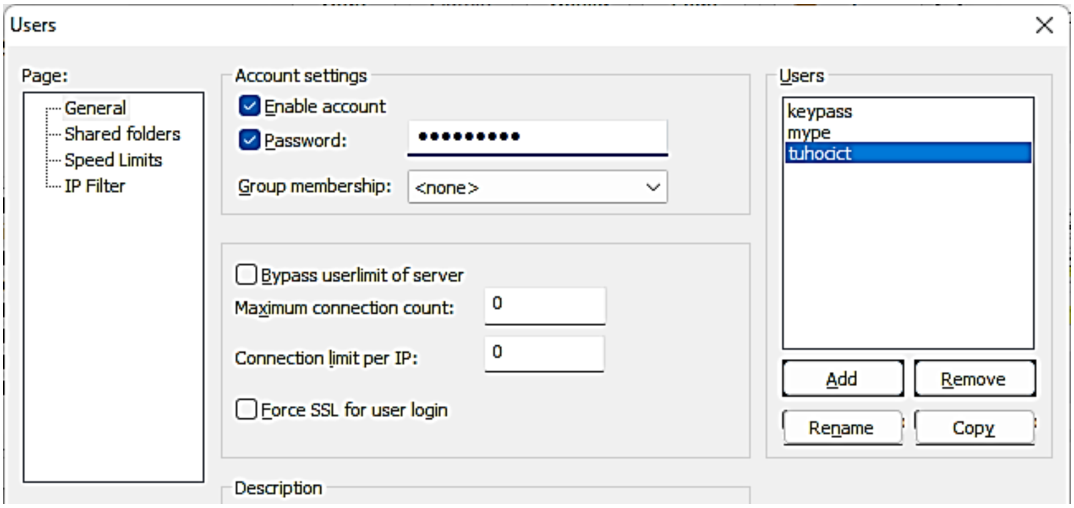
H C:\Users\tuhoc\D...

Files: ☑ Read ☑ Write ☑ Delete ☑ Append

Directories: ☑ Create ☑ Delete ☑ List ☑ + Subdirs

Hình 7.21. Phân quyền người dùng

(5) Vẫn ở cửa sổ Users, click chọn lại node General trong nhóm Page để thiết lập mật khẩu truy cập cho user mới.


Users
Page:
General
Shared folders
Speed Limits
IP Filter
Account settings
Users
keypass
mype
tuhocict
Add
Remove
Rename
Copy
Description

☑ Enable account

☑ Password: •••••••••

Group membership: <none>

☐ Bypass userlimit of server

Maximum connection count: 0

Connection limit per IP: 0

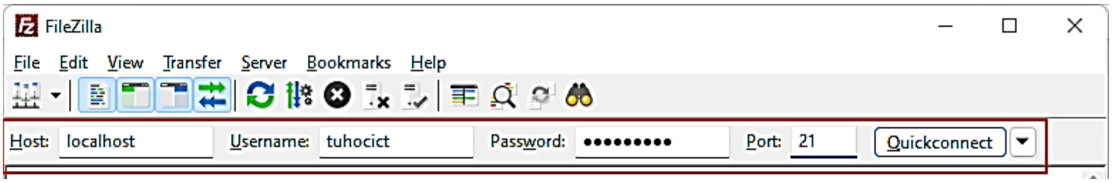
☐ Force SSL for user login

Hình 7.22. Thiết lập mật khẩu cho tài khoản

Đến đây chúng ta đã hoàn thành tạo tài khoản mới cho FileZilla server.

7.3.3. Kết nối FileZilla client với server

Đến đây bạn có thể kết nối client với server. Mở ứng dụng FileZilla client và điền các thông tin kết nối như sau:

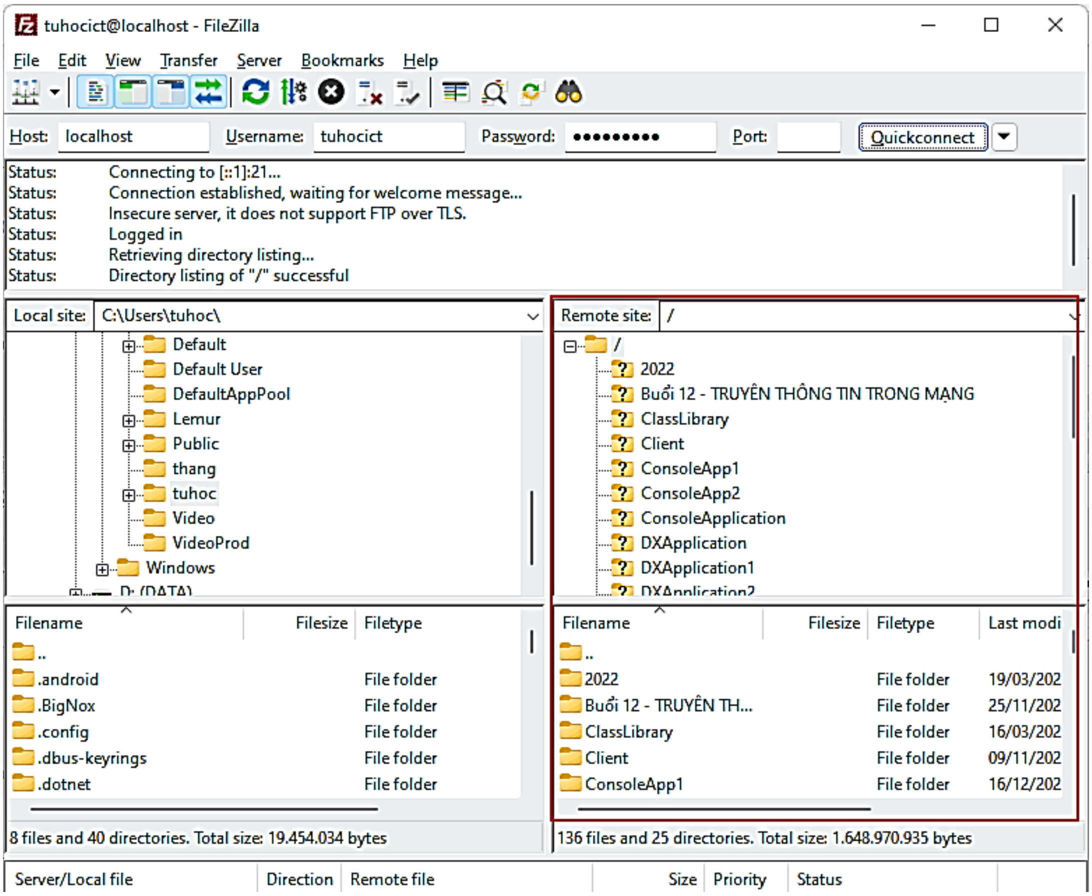

FileZilla
File
Edit
View
Transfer
Server
Bookmarks
Help
Quickconnect

Host: localhost Username: tuhocict Password: ••••••••• Port: 21

Hình 7.23. Cung cấp thông tin kết nối server trên FTP client

Trong đó: Username và Password là tài khoản bạn vừa lập ở bước trên; Port 21 là cổng kênh điều khiển mặc định; Host là localhost do server đang chạy trên máy cục bộ cùng với client. Ấn Quickconnect để kết nối đến server.

Khi kết nối thành công, cửa sổ bên phải sẽ hiển thị cấu trúc thư mục làm việc mà máy chủ cung cấp cho tài khoản, còn cửa sổ bên trái là hệ thống file cục bộ.


tuhocict@localhost - FileZilla
File
Edit
View
Transfer
Server
Bookmarks
Help
Quickconnect

Host: localhost Username: tuhocict Password: ••••••••• Port:

```
Status:     Connecting to [::1]:21...
Status:     Connection established, waiting for welcome message...
Status:     Insecure server, it does not support FTP over TLS.
Status:     Logged in
Status:     Retrieving directory listing...
Status:     Directory listing of "/" successful
```

Local site: C:\Users\tuhoc\

Default
Default User
DefaultAppPool
Lemur
Public
thang
tuhoc
Video
VideoProd
Windows
D: (DATA)

Filename	Filesize	Filetype
..		
.android		File folder
.BigNox		File folder
.config		File folder
.dbus-keyrings		File folder
.dotnet		File folder

8 files and 40 directories. Total size: 19.454.034 bytes

Remote site: /

/
2022
Buổi 12 - TRUYỀN THÔNG TIN TRONG MẠNG
ClassLibrary
Client
ConsoleApp1
ConsoleApp2
ConsoleApplication
DXApplication
DXApplication1
DXApplication2

Filename	Filesize	Filetype	Last modi
..			
2022		File folder	19/03/202
Buổi 12 - TRUYỀN TH...		File folder	25/11/202
ClassLibrary		File folder	16/03/202
Client		File folder	09/11/202
ConsoleApp1		File folder	16/12/202

136 files and 25 directories. Total size: 1.648.970.935 bytes

Server/Local file
Direction
Remote file
Size
Priority
Status

Hình 7.24. Giao diện FileZilla client sau khi kết nối tới server

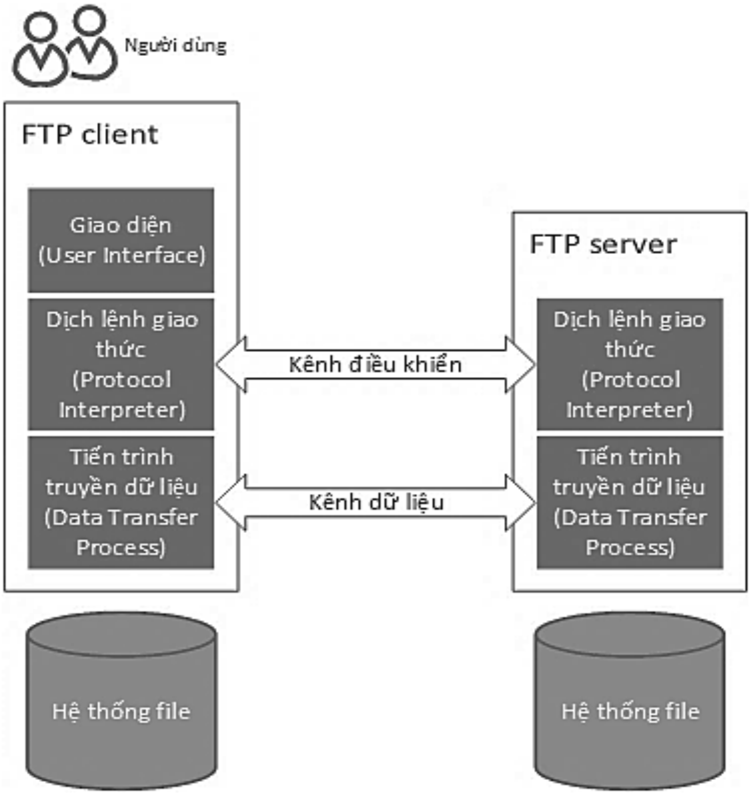
Từ đây bạn có thể làm việc với file và thư mục trên máy chủ giống hệt như đang làm việc với các file và thư mục trên máy cục bộ. Bạn có thể duyệt thư mục trên máy chủ, tạo thư mục mới, xóa thư mục đang có, upload file lên server, download file từ server về, di chuyển file giữa các thư mục, v.v..

7.3.4. Cấu trúc và hoạt động của hệ thống phần mềm FTP

a) Cấu trúc hệ thống phần mềm truyền file

Như ở trên chúng ta đã thấy, hệ thống phần mềm FTP bao gồm hai phần mềm độc lập, FTP client và FTP server. Hai phần mềm này tương tác với nhau sử dụng giao thức FTP.

Cấu trúc chung của hệ thống phần mềm này được trình bày ở Hình 7.25.


Người dùng
FTP client
Giao diện (User Interface)
Dịch lệnh giao thức (Protocol Interpreter)
Tiến trình truyền dữ liệu (Data Transfer Process)
Kênh điều khiển
Kênh dữ liệu
FTP server
Dịch lệnh giao thức (Protocol Interpreter)
Tiến trình truyền dữ liệu (Data Transfer Process)
Hệ thống file
Hệ thống file

Hình 7.25. Cấu trúc hệ thống phần mềm FTP

Phần mềm server bao gồm 2 thành phần: Server Protocol Interpreter (viết tắt la Server PI) và Server Data Transfer Process (viết tắt là Server DTP). Ngoài ra, server phải làm việc được với hệ thống file trên máy chủ.

Phần mềm client bao gồm 3 thành phần: User Protocol Interpreter (User PI), User Data Transfer Process (User DTP), User Interface. Client cũng phải làm việc được với hệ thống file trên máy cục bộ.

Server PI chịu trách nhiệm duy trì một liên kết TCP với User PI. Liên kết TCP này được thiết lập sau khi client xác thực hành công, được duy trì đến khi client đăng xuất, hoặc bị server hủy bỏ (khi client không tương tác quá lâu). Liên kết TCP này được gọi là liên kết điều khiển hoặc kênh điều khiển.

Server DTP chịu trách nhiệm thiết lập liên kết TCP thứ hai với User DTP khi client có yêu cầu truyền tải dữ liệu (tải file về, tải file lên, lấy danh sách file và thư mục). Liên kết này chỉ được thiết lập tạm thời phục vụ truyền dữ liệu. Sau khi hoàn thành truyền dữ liệu, liên kết này sẽ bị hủy bỏ. Liên kết TCP thứ hai này được gọi là liên kết dữ liệu hoặc kênh dữ liệu.

Client có thêm thành phần User Interface. Đây là thành phần giúp người dùng cuối sử dụng phần mềm client. Server không có thành phần tương ứng. User Interface chuyển đổi lệnh của người dùng thành lệnh / nhóm lệnh FTP tương ứng để đưa lên kênh điều khiển. Khi nhận được phản hồi, User Interface chuyển đổi thành dạng người dùng có thể sử dụng. Lệnh của UI và lệnh của FTP là khác nhau.

b) Giao thức FTP

Cấu trúc hệ thống phần mềm FTP ở trên thực thi các quy tắc được quy định bởi giao thức FTP.

Giao thức FTP sử dụng hai kết nối TCP riêng biệt: một kết nối điều khiển (control connection) và một hoặc nhiều kết nối dữ liệu (data connection). Việc phân tách rõ ràng hai loại kênh này giúp quản lý và tối ưu hóa hiệu suất truyền tải.

Kết nối điều khiển (control connection) được thiết lập từ phía client (User PI) đến server (Server PI) và duy trì suốt phiên làm việc FTP. Kết nối này sử dụng cổng TCP 21 mặc định và chịu trách nhiệm truyền các lệnh điều khiển như USER, PASS, LIST, RETR, STOR, v.v.

Khi có yêu cầu truyền dữ liệu (ví dụ như tải file, tải danh sách thư mục), một **kết nối dữ liệu** riêng biệt sẽ được thiết lập giữa client (User DTP) và server (Server DTP). Kết nối này chỉ tồn tại tạm thời trong thời gian thực hiện thao tác truyền tải, sau đó được đóng lại để tiết kiệm tài nguyên.

Giao thức FTP hỗ trợ hai chế độ truyền dữ liệu là chế độ chủ động (active mode) và chế độ thụ động (passive mode), giúp hệ thống linh hoạt thích ứng với các cấu hình mạng khác nhau.

Trong **chế độ chủ động**, client sẽ mở một cổng ngẫu nhiên và lắng nghe kết nối, đồng thời gửi thông tin địa chỉ IP và cổng tới server qua kênh điều khiển. Sau đó, server sử dụng cổng TCP 20 để chủ động kết nối đến client để thực hiện truyền dữ liệu. Tuy nhiên, cách này có thể bị firewall hoặc NAT chặn, vì nó yêu cầu server kết nối ngược trở lại client.

Trong **chế độ thụ động**, server sẽ mở một cổng ngẫu nhiên và gửi thông tin về cổng này cho client. Client sẽ chủ động kết nối đến cổng đó để thiết lập kênh dữ liệu. Đây là chế độ phổ biến hiện nay vì phù hợp với các hệ thống mạng có firewall phía client.

Ví dụ: khi người dùng sử dụng phần mềm FileZilla để tải một tệp tin từ FTP server, quá trình thực hiện sẽ gồm ba bước chính. Đầu tiên, FileZilla thiết lập kênh điều khiển đến cổng 21 của server và tiến hành xác thực đăng nhập. Tiếp theo, khi người dùng gửi lệnh LIST để xem danh sách thư mục, hệ thống sẽ tạo một kết nối dữ liệu tạm thời và nhận dữ liệu từ server. Cuối cùng, khi người dùng gửi lệnh RETR filename.txt để tải file, một kết nối dữ liệu khác lại được thiết lập để truyền nội dung tệp tin.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) Hãy thiết lập một mạng cơ sở (sử dụng cáp xoắn hoặc hotspot). Trên một thiết bị, hãy cài đặt một web server (IIS hoặc Apache) và cấu hình để chấp nhận kết nối từ thiết bị ngoài. Trên thiết bị còn lại chạy trình duyệt và truy cập và web server. Thiết kế một vài trang web đơn giản để thử nghiệm.
2) Tương tự, chạy FileZilla Server và Client trên hai thiết bị kết nối mạng với nhau và thử nghiệm chia sẻ thư mục với nhau.
3) Tìm hiểu cách sử dụng dịch vụ NoIp và cách cấu hình chuyển tiếp cổng trên router để thiết bị từ Internet có thể truy cập vào web server cài đặt trên máy tính cục bộ tại gia đình.

Câu hỏi lý thuyết

1) Tại sao giao thức FTP lại sử dụng hai kết nối TCP riêng biệt (điều khiển và dữ liệu), trong khi HTTP chỉ dùng một kết nối? Điều này mang lại lợi ích gì?
2) Giả sử bạn muốn tải nhiều tệp cùng lúc từ một server FTP, cách hoạt động của FTP sẽ ảnh hưởng như thế nào đến hiệu suất và độ trễ?
3) Nếu một địa chỉ IP có thể trỏ tới nhiều tên miền khác nhau, hệ thống DNS phải xử lý thế nào để phân giải chính xác? Việc này mang lại lợi ích gì cho doanh nghiệp?
4) Tại sao người dùng thường truy cập website qua tên miền thay vì địa chỉ IP, và điều gì sẽ xảy ra nếu dịch vụ DNS tạm thời không hoạt động?
5) Trong cùng một thời điểm, tại sao bạn không thể chạy song song cả Apache và IIS trên cổng 80 nếu không thay đổi cấu hình? Điều này liên quan đến nguyên lý nào trong hệ điều hành và mạng?
6) Khi một trình duyệt truy cập địa chỉ http://localhost/index.html, các thành phần nào trong hệ thống phần mềm Web sẽ tham gia vào quá trình xử lý và phản hồi?
7) Tại sao các chế độ truyền dữ liệu chủ động (active) và thụ động (passive) lại cần thiết trong FTP? Mỗi chế độ có ưu nhược điểm gì khi triển khai thực tế?
8) Giả sử bạn thay đổi nội dung một file HTML trong thư mục của Apache, nhưng trình duyệt vẫn hiển thị nội dung cũ. Nguyên nhân nào có thể gây ra vấn đề này?

BÀI 8. LẬP TRÌNH MẠNG

Mục tiêu của bài học

Bài học giúp sinh viên nắm vững kiến thức và kỹ năng cơ bản trong việc phát triển ứng dụng mạng bằng lập trình socket, thiết kế giao thức tầng ứng dụng và triển khai mô hình giao tiếp phân tán. Nội dung bài học hướng đến việc kết hợp giữa lý thuyết giao thức và thực hành xây dựng phần mềm mạng, đặc biệt với các ví dụ cụ thể sử dụng C# và giao thức HTTP, ICMP.

Về kiến thức:

- Trình bày được vai trò của socket và API trong việc phát triển ứng dụng mạng.
- Mô tả được các mô hình kiến trúc ứng dụng mạng: client/server, peer-to-peer.
- Phân biệt được các mô hình giao tiếp như request/response, publish/subscribe và pipeline.
- Hiểu rõ cấu trúc gói tin HTTP và ICMP, cũng như cơ chế hoạt động của các ứng dụng mạng phổ biến.

Về kỹ năng:

- Xây dựng các ứng dụng mạng đơn giản sử dụng TCP socket với ngôn ngữ C#.
- Phát triển ứng dụng client/server tương tác qua giao thức HTTP.
- Lập trình ứng dụng ping sử dụng socket thô và phân tích gói tin ICMP.
- Thiết kế và triển khai giao thức tầng ứng dụng cho hệ thống phần mềm phân tán.

8.1. Xây dựng ứng dụng mạng và giao thức tầng ứng dụng

8.1.1. Phát triển phần mềm mạng

Phát triển phần mềm ứng dụng mạng, về bản chất, là viết một hoặc nhiều chương trình chạy trên các thiết bị đầu cuối khác nhau và giao tiếp với nhau qua mạng.

a) Lập trình ứng dụng mạng

Ứng dụng mạng có thể được viết bằng các ngôn ngữ lập trình thông dụng như C/C++, Java, Python, C#, Visual Basic v.v., tương tự như bất kỳ một ứng dụng "không mạng" nào. Các ngôn ngữ và công cụ lập trình cấp cao còn cung cấp những hỗ trợ bổ sung để tăng hiệu quả của việc phát triển ứng dụng mạng.

Trong giáo trình này chúng ta sẽ sử dụng ngôn ngữ C# và nền tảng .NET để minh họa cho việc phát triển ứng dụng mạng. Tuy nhiên, giáo viên và sinh viên có thể sử dụng bất kỳ ngôn ngữ lập trình nào để thực hiện các ví dụ này. Kỹ thuật lập trình socket không khác biệt quá nhiều giữa các ngôn ngữ.

Khi lập trình ứng dụng mạng, lập trình viên phải giao tiếp được với ngăn xếp giao thức mạng do hệ điều hành cung cấp để yêu cầu các dịch vụ này thực hiện việc gửi và nhận dữ liệu qua mạng.

Tất cả các hệ điều hành hiện đại đều cung cấp cho ứng dụng khả năng sử dụng các dịch vụ truyền thông mạng. Các hệ điều hành hiện đại cung cấp các dịch vụ truyền thông dưới dạng giao diện lập trình ứng dụng API (Application Programming Interface), gọi là **socket API**. Sử dụng socket API là cách thức cơ bản để phần mềm ứng dụng tương tác với dịch vụ truyền thông của hệ điều hành.

Ngoài việc trực tiếp sử dụng socket API, các framework dành cho phát triển ứng dụng còn hỗ trợ khai thác dịch vụ truyền thông mạng mà người lập trình thậm chí không cần phải nắm rõ hoạt động của mạng máy tính.

b) Ứng dụng mạng và giao thức tầng giao vận

Mỗi ứng dụng mạng gần như bắt buộc phải sử dụng đến dịch vụ giao vận (dịch vụ truyền thông end-to-end) do hai giao thức tầng giao vận (TCP và UDP) cung cấp. Chương trình ứng dụng trực tiếp khai thác một trong hai giao thức này để truyền dữ liệu. Việc khai thác dịch vụ truyền TCP hoặc UDP được thực hiện thông qua khai thác Socket API tương ứng.

Tùy thuộc vào dịch vụ giao vận được lựa chọn, người lập trình phải tuân thủ theo quy trình hoạt động của giao thức đó.

Ví dụ, nếu lựa chọn dịch vụ TCP, chúng ta phải tuân thủ quy tắc hoạt động của TCP, bao gồm: thiết lập liên kết, gửi/nhận dữ liệu, đóng liên kết; Nếu lựa chọn dịch vụ UDP thì có thể trực tiếp gửi/nhận dữ liệu giữa các điểm cuối mà không cần thiết lập liên kết.

Dịch vụ tầng giao vận sẽ thực hiện các yêu cầu và người lập trình không cần quan tâm đến cách thức chúng thực hiện yêu cầu đó.

c) Ứng dụng mạng và giao thức tầng ứng dụng

Như chúng ta đã xem xét ở phần trên, các phần của ứng dụng mạng trao đổi thông tin với nhau bằng cách gọi một trong số các dịch vụ truyền dữ liệu của tầng giao vận.

Tuy nhiên, rất nhanh sẽ xuất hiện các vấn đề như: cấu trúc các gói tin ra sao, ý nghĩa của từng trường trong gói tin, khi nào các chương trình thành phần gửi và nhận các gói tin, v.v.. Trả lời cho những câu hỏi này có thể xem gần tương đương như việc xây dựng một giao thức tầng ứng dụng.

Cụ thể hơn, một giao thức tầng ứng dụng xác định:

- Các loại thông điệp cần trao đổi: ví dụ, hai loại thông điệp thường gặp nhất là truy vấn (request) và phản hồi (response).
- Cú pháp của các loại thông điệp: ví dụ, danh sách các trường trong từng loại thông điệp và cách thức tạo ra các trường đó.
- Ý nghĩa của các trường: giá vị và vai trò của thông tin trong các trường.

- Các quy tắc xác định thời điểm và cách thức một thành phần gửi thông điệp và phản hồi cho thông điệp.

Điều này có nghĩa là, khi phát triển ứng dụng mạng, dù muốn hay không chúng ta cũng phải xây dựng một giao thức tầng ứng dụng. Nếu không có khả năng tự xây dựng giao thức, chúng ta phải sử dụng đến những giao thức sẵn có.

Các giao thức tầng ứng dụng rất đa dạng. Ngoài những giao thức mở thông dụng, còn rất nhiều giao thức tầng ứng dụng khác do người lập trình tự thiết kế và thực thi.

Các chương trình ứng dụng mạng bắt buộc phải tự mình xây dựng các giao thức riêng (giao thức độc quyền) hoặc sử dụng một giao thức sẵn có (giao thức mở). Các giao thức này đều thuộc về tầng ứng dụng. Do vậy có rất nhiều giao thức tầng ứng dụng khác nhau.

Các giao thức tầng ứng dụng luôn đi kèm với chương trình và thường được tích hợp thành một module của chương trình. Ví dụ, giao thức HTTP luôn đi kèm với hệ thống phần mềm web. Nó được tích hợp trong chương trình trình duyệt và chương trình chủ web. Một số giao thức tầng ứng dụng cũng được xây dựng ở dạng giao thức "đa năng", nghĩa là nó có thể được nhúng vào và sử dụng bởi bất kỳ phần mềm nào.

d) Công nghệ hỗ trợ phát triển ứng dụng mạng

Để đơn giản hóa việc phát triển ứng dụng mạng, nhiều công nghệ lập trình hỗ trợ làm việc với các giao thức tầng ứng dụng phổ biến như HTTP, POP, SMTP, IMAP, FTP. Các công cụ cấp cao này hoàn toàn ẩn đi các chi tiết liên quan tới lập trình mạng cấp thấp.

Khi sử dụng các công cụ này, bạn được cung cấp sẵn các công cụ để thực hiện việc truyền dữ liệu giữa chương trình khách và chủ qua giao thức HTTP. Bạn sẽ không cần quan tâm đến cách thức truyền dữ liệu (cũng không cần quan tâm đến HTTP) nữa mà chỉ cần quan tâm đến bản thân dữ liệu đó thôi.

Hầu hết các công nghệ/công cụ hỗ trợ phát triển ứng dụng cấp cao đều hỗ trợ đầy đủ việc truyền dữ liệu giữa chương trình khách và chủ qua HTTP. Trên thực tế HTTP là một sự lựa chọn đặc biệt phổ biến hiện nay để phát triển các ứng dụng mạng, nhất là các ứng dụng quản lý, đồng thời nó cũng được các công nghệ phát triển ứng dụng hỗ trợ rất tốt.

HTTP không phải là giải pháp tốt cho nhiều loại ứng dụng. Ví dụ, các ứng dụng trò chơi trực tuyến không thể sử dụng HTTP để trao đổi dữ liệu vì HTTP rất chậm và không duy trì trạng thái qua mỗi phiên truy vấn / phản hồi. Trong những trường hợp này, bạn cần tự thiết kế giao thức ứng dụng riêng và phải sử dụng kỹ thuật lập trình socket để thực thi giao thức.

8.1.2. Mô hình trao đổi thông điệp trong ứng dụng mạng

Trong một ứng dụng mạng, bên cạnh sự phân chia theo chức năng nhiệm vụ, một vấn đề quan trọng khác cũng quyết định kiến trúc của hệ thống, đó là mô hình trao đổi thông điệp.

a) Khái niệm mô hình trao đổi thông điệp

Mô hình trao đổi thông điệp (Messaging Pattern, Message Exchange Pattern), cũng được gọi là mô hình giao tiếp, là một mẫu kiến trúc mô tả cách kết nối và tương tác giữa các thành phần khác nhau của một hệ thống phần mềm phân tán, và xác định cách thức thông điệp được trao đổi trong một hệ thống phần mềm phân tán.

Mô hình trao đổi thông điệp giúp tách biệt các thành phần của hệ thống, cho phép chúng hoạt động độc lập với nhau. Điều này giúp giảm độ phức tạp của hệ thống, dễ dàng mở rộng và bảo trì. Nó cũng cho phép giao tiếp giữa các ứng dụng chạy trên các nền tảng khác nhau (máy chủ, máy tính cá nhân, điện thoại di động và thiết bị IoT) và được phát triển bởi các ngôn ngữ lập trình khác nhau.

Người ta cũng xây dựng nhiều giao thức trao đổi thông điệp. Có thể hình dung các giao thức trao đổi thông điệp là những giao thức ở tầng ứng dụng nhưng "đa năng", thay vì chỉ phục vụ mục đích riêng của loại ứng dụng đó. Các giao thức trao đổi thông điệp phổ biến trong ứng dụng mạng bao gồm ZeroMQ, MQTT, AMQP.

Các giao thức trao đổi thông điệp đều hỗ trợ một hoặc một số mô hình trao đổi thông điệp. Có rất nhiều mô hình trao đổi thông điệp khác nhau. Ở đây chúng ta chỉ xem xét một số mô hình phổ biến và được ứng dụng nhiều.

b) Mô hình truy vấn phản hồi

Mô hình truy vấn phản hồi cũng được gọi là mô hình request / response hoặc mô hình request / reply.

Trong mô hình truy vấn / phản hồi, có hai thành phần chính:

- Thành phần yêu cầu (requester): Thành phần này gửi yêu cầu và chờ đợi phản hồi tương ứng.
- Thành phần phản hồi (responder): Thành phần này chuyên nhận yêu cầu và phản hồi tương ứng.

Trong mô hình này, các thành phần yêu cầu không trao đổi thông tin trực tiếp với nhau. Thành phần yêu cầu chỉ trao đổi thông tin với thành phần phản hồi. Do đó, nó tạo ra kiểu trao đổi thông tin 1 - 1 (unicast) truyền hai chiều không đồng thời (half-duplex).

Đây là mô hình trao đổi thông điệp rất phổ biến trong kiến trúc chủ / khách. Trong kiến trúc này, khách thường đóng vai trò yêu cầu, và chủ đóng vai trò phản hồi.

Mô hình truy vấn phản hồi được sử dụng phổ biến trong các giao thức "cổ điển" như HTTP, FTP, các giao thức cho thư điện tử (SMTP, POP).

c) Mô hình publish / subscribe

Mô hình xuất bản/đăng ký, còn được gọi tắt là mô hình pub/sub hoặc mô hình xuất bản/đăng ký, được phân thành hai nhóm chính trong hệ thống:

- **Người xuất bản** (Publisher): Đây là thành phần tạo ra các thông điệp và gửi chúng tới một trung tâm trung gian, cũng được gọi là message broker. Mỗi thông điệp được liên kết với một chủ đề cụ thể. Người xuất bản không biết ai sẽ nhận được thông điệp của mình, và trung tâm trung gian sẽ chuyển tiếp thông điệp đến các người đăng ký khi có yêu cầu theo chủ đề tương ứng.

- **Người đăng ký** (Subscriber): Đây là thành phần nhận thông điệp từ trung tâm trung gian. Người đăng ký chỉ nhận các thông điệp theo chủ đề mà nó đã đăng ký.

Trong mô hình xuất bản/đăng ký, các thông điệp được phân phối đến người đăng ký cùng một lúc và không đồng bộ. Điều này đảm bảo rằng mỗi người đăng ký chỉ nhận các thông điệp mà họ quan tâm và không cần xử lý các thông điệp không cần thiết. Mô hình này cũng cho phép các thành phần trong hệ thống hoạt động độc lập và không phụ thuộc vào sự hiện diện của nhau. Như vậy, mô hình xuất bản/đăng ký tạo ra một kiểu trao đổi thông tin 1-nhiều (multicast/broadcast) giữa một người xuất bản và nhiều người đăng ký.

Mô hình xuất bản/đăng ký có hai kiểu triển khai. Kiểu đầu tiên sử dụng một trung tâm trung gian riêng biệt (broker). Kiểu thứ hai sử dụng người xuất bản làm trung tâm trung gian.

Mô hình xuất bản/đăng ký được sử dụng rộng rãi trong các ứng dụng IoT. Khi áp dụng mô hình này trong phần mềm, thường sử dụng cơ chế bất đồng bộ và hướng sự kiện. Mô hình này cũng cho phép mỗi phần mềm trong hệ thống có khả năng đồng thời gửi (publish) và nhận (subscribe) dữ liệu. Do đó, mô hình này rất phù hợp cho việc điều khiển các thiết bị và thu thập dữ liệu từ các thiết bị phân tán.

d) Mô hình pipeline

Mô hình pipeline, hay còn được gọi là mô hình push/pull, là một mô hình giao tiếp trong đó có hai loại ứng dụng chính: pusher (người đẩy) và puller (người kéo). Pusher là người gửi các thông điệp thông qua đường ống, trong khi puller là người nhận các thông điệp đó. Mỗi pusher có khả năng gửi thông điệp đến nhiều puller, và mỗi puller cũng có thể nhận thông điệp từ nhiều pusher.

Trong mô hình này, các thông điệp được truyền qua đường ống theo trình tự gửi ban đầu. Tuy nhiên, các giao thức thường không đảm bảo rằng các thông điệp sẽ được nhận theo đúng thứ tự gửi. Vì vậy, các ứng dụng phải tự quản lý việc sắp xếp và xử lý các thông điệp theo đúng thứ tự mong muốn.

Mô hình giao tiếp pipeline thường được áp dụng trong các hệ thống xử lý dữ liệu lớn, khi các ứng dụng cần thực hiện xử lý dữ liệu theo một quy trình liên tục, từng bước một. Khi một bước xử lý hoàn thành, kết quả của nó được truyền đến bước tiếp theo để tiếp tục xử lý. Điều này giúp tăng tốc độ xử lý và linh hoạt trong việc xử lý dữ liệu theo luồng công việc cụ thể.

Mô hình pipeline cũng cho phép các ứng dụng hoạt động độc lập và có khả năng mở rộng. Các pusher và puller có thể tồn tại trên các thiết bị và môi trường khác nhau, và việc truyền thông điệp giữa chúng đơn giản và hiệu quả. Điều này đặc biệt hữu ích trong các môi trường phân tán và các hệ thống có khối lượng dữ liệu lớn.

8.1.3. Kiến trúc ứng dụng mạng

Khi phát triển phần mềm mạng, một vấn đề cần giải quyết ngay từ đầu là sự thỏa thuận về mặt nhiệm vụ giữa các thành phần trong hệ thống. Sự thỏa thuận đó thể hiện qua việc lựa chọn mô hình ứng dụng mạng.

Mô hình ứng dụng mạng, cũng gọi là kiến trúc ứng dụng mạng, là sự phân chia vai trò nhiệm vụ của các thành phần trong một hệ thống phần mềm ứng dụng mạng, và cách thức các thành phần này tương tác với nhau.

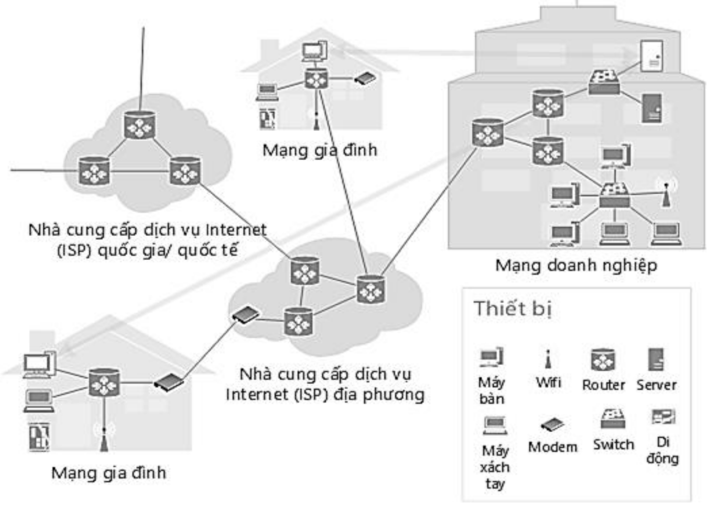
Mặc dù phần mềm mạng rất đa dạng nhưng có thể phân chia chúng vào hai mô hình cơ bản:

- Mô hình chủ / khách (client / server)
- Mô hình ngang hàng (peer-to-peer, P2P)

a) Mô hình chủ / khách

Trong mô hình chủ / khách (Hình 8.1), ứng dụng được chia làm hai thành phần: một thành phần chuyên phục vụ các yêu cầu gửi đến từ các thành phần khác, gọi là chương trình chủ (server); một hoặc nhiều thành phần đưa ra yêu cầu sử dụng dịch vụ, gọi là chương trình khách (client).

Chương trình chủ và khách chạy trên các thiết bị đầu cuối khác nhau khác nhau. Thiết bị đầu cuối nơi chạy chương trình chủ cũng thường được gọi tắt là máy chủ. Thiết bị đầu cuối nơi chạy chương trình khách thường được gọi tắt là máy khách hay máy trạm. Trên máy chủ vật lý cũng thường cài đặt nhiều chương trình chủ khác nhau.


Mạng gia đình
Nhà cung cấp dịch vụ Internet (ISP) quốc gia/ quốc tế
Mạng doanh nghiệp
Thiết bị
Máy bàn
Wifi
Router
Server
Máy xách tay
Modem
Switch
Di động
Nhà cung cấp dịch vụ Internet (ISP) địa phương
Mạng gia đình

Hình 8.1. Mô hình chủ khách

Ví dụ, hệ thống phần mềm web bao gồm một chương trình luôn hoạt động (chương trình chủ web) trên một máy tính riêng để chờ phục vụ các truy vấn đến từ các chương trình trình duyệt (chương trình khách - browser). Khi chương trình chủ nhận được một truy vấn từ trình duyệt, nó sẽ xử lý và trả lại dữ liệu mà trình duyệt yêu cầu. Các trình duyệt chỉ trao đổi thông tin với chương trình máy chủ mà không tương tác trực tiếp với nhau. Một số ứng dụng phổ biến khác sử dụng mô hình chủ-khách khác bao gồm phần mềm truyền file FTP, phần mềm Email.

Mô hình chủ khách có các đặc điểm quan trọng sau:

- Chương trình chủ luôn hoạt động để chờ phục vụ khách. Ví dụ, chương trình máy chủ web phải hoạt động toàn thời gian để có thể phục vụ trình duyệt bất kỳ lúc nào. Chế độ hoạt động như vậy gọi là chế độ luôn mở.
- Chương trình khách có thể hoạt động ở chế độ luôn mở (always-on) hoặc mở không thường xuyên (sometimes-on). Ví dụ, trong hệ thống web, trình duyệt có thể hoạt động hoặc có thể không (tức là chế độ mở không thường xuyên).
- Chương trình chủ phải hoạt động trên máy có địa chỉ IP cố định mà chương trình khách có thể luôn xác định được. Địa chỉ này cũng phải công bố để chương trình khách biết. Một loại địa chỉ khác thường được dùng là tên miền (domain name). Ví dụ, địa chỉ IP của máy chủ báo Dân Trí là 183.81.34.136, còn tên miền tương ứng là dantri.com.vn.
- Chương trình khách phải biết được địa chỉ của máy chủ. Ví dụ, nếu muốn xem nội dung trang web nào, chúng ta phải cung cấp địa chỉ của nó (ở dạng tên miền hoặc địa chỉ IP).
- Các chương trình khách không giao tiếp trực tiếp với nhau mà chỉ giao tiếp với chương trình chủ. Ví dụ, hai trình duyệt không bao giờ gửi dữ liệu trực tiếp cho nhau mà đều phải thông qua chương trình chủ web.

Việc giao tiếp giữa chương trình khách và chủ được thực hiện dưới hình thức trao đổi các thông điệp (message):

- Khách gửi thông điệp yêu cầu (request message) cho chương trình chủ để mô tả công việc cần thực hiện.
- Khi nhận được thông điệp này, chương trình chủ sẽ phân tích và xác định công việc phải thực hiện.
- Nếu quá trình xử lý sinh ra kết quả cần trả cho client thì chương trình chủ sẽ gửi cho chương trình khách thông điệp trả lời (reply message).
- Định dạng và ý nghĩa của các thông điệp trao đổi giữa chương trình chủ và khách được quy định bởi giao thức dùng trong ứng dụng.

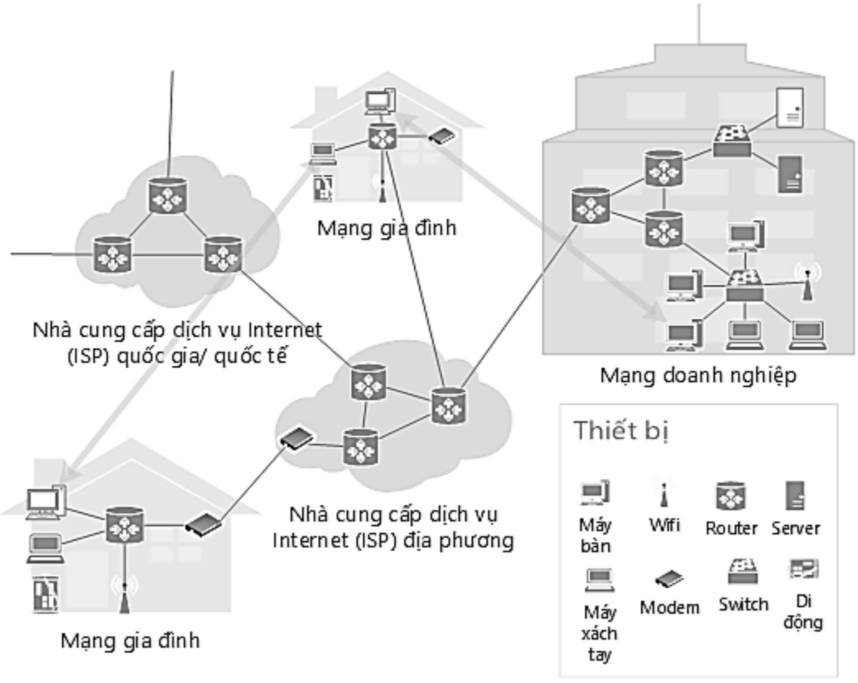
Trong mô hình chủ khách, tùy thuộc vào việc phân phối nhiệm vụ giữa chủ và khách, người ta còn phân chia làm hai loại chương trình khách:

- Fat Client: Trong mô hình Fat Client, chương trình chủ chịu trách nhiệm truy vấn và lưu trữ thông tin. Mô hình này tạo ra ít lưu thông trên mạng do dữ liệu được lưu trữ tạm thời và tính toán ở chương trình khách nhưng yêu cầu cấu hình máy khách cao hơn.
- Thin Client: Trong mô hình Thin Client, chương trình khách chỉ cần hiển thị và kiểm tra tính hợp lệ của dữ liệu người dùng nhập vào. Mô hình này cho phép tạo ra các chương trình khách nhẹ và đơn giản nhưng làm tăng lưu thông trên mạng do dữ liệu phải chuyển qua lại liên tục giữa máy chủ và khách.

Trong ứng dụng chủ/khách, thường một máy chủ vật lý đơn lẻ không đủ khả năng để xử lý tất cả các truy vấn từ các chương trình khách. Ví dụ, một trang web mạng xã hội quy mô lớn sẽ dễ dàng gặp tình trạng quá tải nếu chỉ dựa trên một máy chủ duy nhất để phục vụ. Vì lý do này, các trung tâm dữ liệu (data center) đã được phát triển, chứa hàng ngàn máy chủ vật lý, nhằm tạo ra các máy chủ ảo mạnh mẽ hơn. Ví dụ, Google, theo thống kê từ Wikipedia, sở hữu từ 30 đến 50 trung tâm dữ liệu trên toàn cầu, với hơn hai triệu máy chủ được sử dụng để hỗ trợ các yêu cầu tìm kiếm, dịch vụ Gmail và YouTube. Mỗi trung tâm dữ liệu có chứa hàng trăm ngàn máy chủ vật lý.

b) Mô hình ngang hàng

Trong kiến trúc ngang hàng (còn gọi là kiến trúc peer-to-peer hay viết tắt là P2P), tất cả các thành phần của hệ thống đều thực hiện các nhiệm vụ giống nhau, không có thành phần phục vụ (server) chuyên biệt, không có thành phần yêu cầu (client) chuyên biệt (Hình 8.2).


Mạng gia đình
Nhà cung cấp dịch vụ Internet (ISP) quốc gia/ quốc tế
Mạng doanh nghiệp
Nhà cung cấp dịch vụ Internet (ISP) địa phương
Mạng gia đình
Thiết bị
Máy bàn
Wifi
Router
Server
Máy xách tay
Modem
Switch
Di động

Hình 8.2. Mô hình peer-to-peer

Trong ứng dụng P2P không có sự phụ thuộc (hoặc phụ thuộc rất ít) vào chương trình chủ. Mỗi ứng dụng chạy trên một máy gọi là peer. Ứng dụng trên mỗi đôi máy (các peer) trực tiếp tiến hành truyền thông với nhau mà không phải thông qua một chương trình chủ trung gian nào. Vì lý do này mô hình được gọi là peer-to-peer. Các peer này không do các nhà cung cấp dịch vụ quản lý mà thực chất là các máy tính cá nhân do người dùng quản lý.

Hiện nay các ứng dụng đòi hỏi lưu lượng dữ liệu lớn thường hoạt động theo mô hình P2P: ứng dụng phân phối file (BitTorrent, µTorrent), ứng dụng chia sẻ file (eMule, LimeWire), điện thoại Internet (Skype), truyền hình IP (PPLive).

Ưu điểm của P2P:

- Kiến trúc P2P có khả năng mở rộng. Ví dụ, trong ứng dụng chia sẻ file P2P, mặc dù mỗi peer trong khi yêu cầu file tự tạo ra một khối lượng công việc cho mình, mỗi peer đồng thời bổ sung khả năng phục vụ của mình cho hệ thống bằng cách phân phối lại file cho các peer khác.
- Kiến trúc P2P cũng có ưu thế về giá thành vì nó không đòi hỏi hạ tầng về máy chủ và băng thông.

Nhược điểm của P2P:

- Các nhà cung cấp dịch vụ Internet (ISP) cho hộ gia đình thường để tốc độ tải xuống lớn hơn tốc độ tải lên. Trong khi đó, dịch vụ phát video (streaming) hoặc phân phối file P2P đều đòi hỏi tốc độ tải lên cao. Từ đó, ISP địa phương phải chịu áp lực lớn về băng thông. Người ta gọi đây là tình trạng "không thân thiện" giữa ứng dụng P2P và ISP.
- Do bản chất mở và phân tán, ứng dụng P2P có hạn chế về tính an toàn và bảo mật.
- Ứng dụng P2P cũng dễ dàng liên quan tới các vấn đề vi phạm bản quyền vì dữ liệu chia sẻ không bị kiểm soát.

Sự thành công của ứng dụng P2P phụ thuộc vào khả năng "thuyết phục" người dùng tham gia vào mạng lưới ứng dụng P2P. Khi tham gia, người dùng sẽ phải "tự nguyện" cung cấp tài nguyên của máy tính cá nhân cho ứng dụng như băng thông, lưu trữ, CPU. Ví dụ, khi tham gia vào hệ thống chia sẻ file P2P, mỗi phần mềm trên máy tính cá nhân đều phải chiếm dụng thêm băng thông, ổ cứng, thời gian xử lý của CPU và cung cấp tài nguyên này cho hệ thống cùng sử dụng.

Ngoài hai mô hình trên, hiện nay một số ứng dụng sử dụng kiến trúc lai, kết hợp cả mô hình chủ-khách và mô hình P2P. Ví dụ, trong nhiều ứng dụng nhắn tin tức thời, máy chủ chỉ dùng để theo dõi địa chỉ IP của người dùng, còn tin nhắn giữa người dùng được truyền trực tiếp giữa các máy khách mà không đi qua máy chủ.

8.2. Lập trình với giao thức HTTP

8.2.1. Thực hành - Xây dựng web client đơn giản

Trong bài thực hành này, chúng ta sẽ cùng xây dựng một web client đơn giản nhất. Web client này chỉ có thể gửi một truy vấn về server. Khi nhận phản hồi nó sẽ in kết quả ra giao diện console.

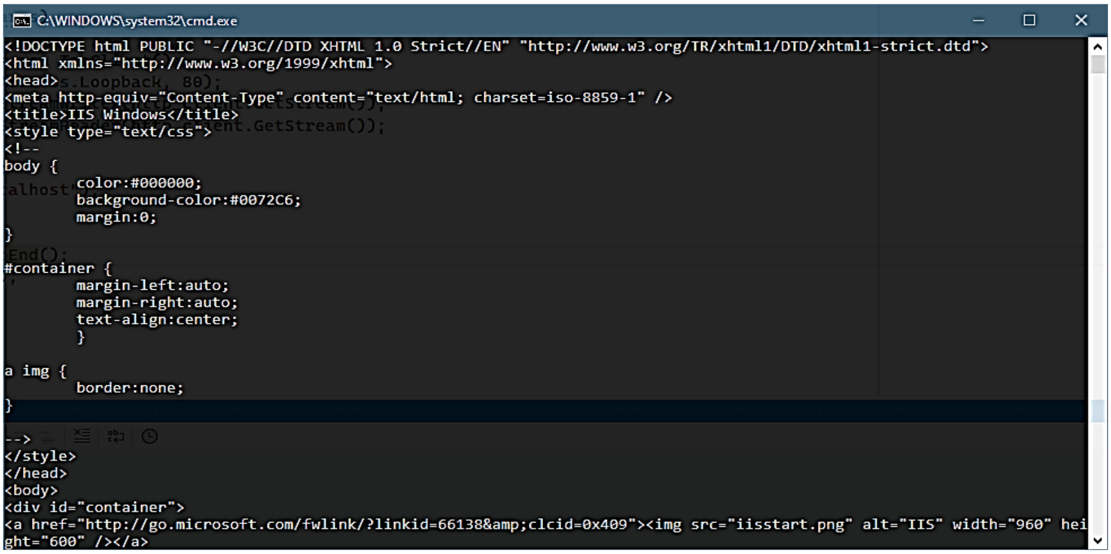
(1) Tạo dự án C# (Console App) và đặt tên cho dự án là HttpClient_Console.

(2) Mở file Program.cs và viết code như sau:

Program.cs

```
using System;
using System.Net;
using System.Net.Sockets;
using System.IO;
namespace HttpClient;
class Program {
    static void Main(string[] args) {
        TcpClient http_client = new TcpClient();
        http_client.Connect(IPAddress.Loopback, 80);
        StreamWriter sw = new StreamWriter(http_client.GetStream());
        var request = "GET /\r\nHost: localhost";
        sw.WriteLine(request);
        sw.Flush();
        StreamReader sr = new StreamReader(http_client.GetStream());
        var response = sr.ReadToEnd();
        Console.WriteLine(response);
        sw.Close();
        sr.Close();
        Console.ReadKey();
    }
}
```

(3) Dịch và chạy chương trình. Giả sử bạn đang sử dụng server IIS, kết quả sẽ như sau:



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS Windows</title>
<style type="text/css">
<!--
body {
        color:#000000;
        background-color:#0072C6;
        margin:0;
}

#container {
        margin-left:auto;
        margin-right:auto;
        text-align:center;
        }

a img {
        border:none;
}

-->
</style>
</head>
<body>
<div id="container">
<a href="http://go.microsoft.com/fwlink/?linkid=66138&amp;clcid=0x409"><img src="iisstart.png" alt="IIS" width="960" hei
ght="600" /></a>
```

Hình 8.3. Kết quả chạy chương trình HTTP client đơn giản

Đây là mã HTML mà chương trình chủ IIS gửi lại cho chương trình qua HTTP. Nếu sử dụng Apache, kết quả thu được có thể sẽ khác.

Trong ví dụ trên, chúng ta đã thực hiện các thao tác sau:

(1) Thiết lập một liên kết TCP tới cổng số 80 của máy tính cục bộ:

```
TcpClient http_client = new TcpClient();
http_client.Connect(IPAddress.Loopback, 80);
```

(2) Trên liên kết TCP này, chúng ta gửi đi một chuỗi văn bản theo cấu trúc của HTTP request:

```
StreamWriter sw = new StreamWriter(http_client.GetStream());
var request = "GET /\r\nHost: localhost";
sw.WriteLine(request);
sw.Flush();
```

Ở đây chúng ta sử dụng Stream Adapter StreamWriter để tiện lợi cho việc gửi từng dòng văn bản lên luồng mạng TCP.

(3) Sau khi gửi đi chuỗi truy vấn, chúng ta đọc toàn bộ phản hồi về chuỗi html và in ra màn hình console:

```
StreamReader sr = new StreamReader(http_client.GetStream());
var response = sr.ReadToEnd();
Console.WriteLine(response);
```

Ở đây, chúng ta sử dụng Stream Adapter StreamReader để tiện lợi cho việc đọc cả chuỗi văn bản lớn từ luồng mạng TCP.

Như vậy, chúng ta có thể thấy, chương trình do chúng ta viết đã có thể trao đổi thông tin với web server IIS / Apache thông qua giao thức HTTP.

Đây là ví dụ đơn giản nhất về việc tạo gói tin truy vấn HTTP và nhận phản hồi từ server.

Lưu ý, mặc dù bạn có thể tải nội dung trang web HTML về ứng dụng, bạn không thể hiển thị nó như trong trình duyệt. HTML bản chất chỉ là chuỗi ký tự và nó cần các cơ chế của trình duyệt để có thể hiển thị thành giao diện đồ họa.

8.2.2. Thực hành - Xây dựng web server đơn giản

Trong phần này, chúng ta sẽ cùng xây dựng một web server đơn giản. Web server này có thể nhận truy vấn từ trình duyệt thực sự (Chrome, Edge, Firefox, v.v.) và trả về một đoạn HTML.

(1) Tạo một dự án mới kiểu Console App, đặt tên là HttpServerSimple.

(2) Mở file Program.cs và viết code như sau:

Program.cs

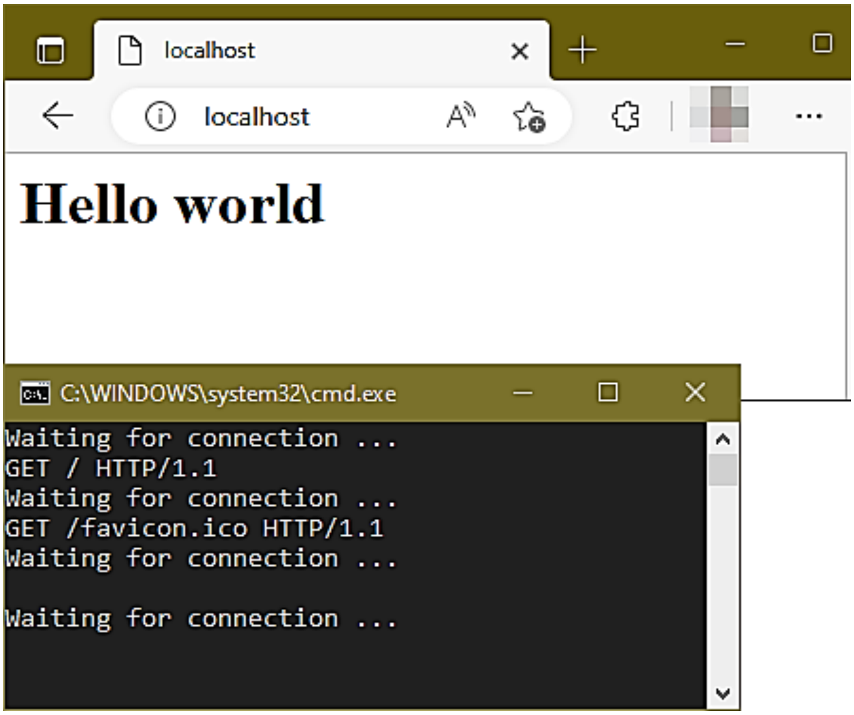
```
using System;
using System.Net;
using System.Net.Sockets;
using System.IO;
namespace HttpServerSimple;
class Program {
    static void Main(string[] args) {
        TcpListener listener = new TcpListener(IPAddress.Any, 80);
        listener.Start(5);
        while (true) {
            try {
                Console.WriteLine("Waiting for connection ...");
                var client = listener.AcceptTcpClient();
                var sr = new StreamReader(client.GetStream());

                string request = sr.ReadLine();
                Console.WriteLine(request);
                var sw = new StreamWriter(client.GetStream());
                sw.WriteLine("HTTP/1.1 200 OK\n");
                sw.WriteLine("<html><head></head><body><h1>Hello
world</h1><body></html>");
                sw.Flush();
                client.Close();
            } catch (Exception) { }
        }
    }
}
```

(3) Kiểm tra xem IIS hay Apache có hoạt động hay không. Nếu các chương trình này hoạt động thì cần tắt đi.

(4) Dịch và chạy thử chương trình.

(5) Mở một trình duyệt bất kỳ và gõ vào đường link: http://localhost. Bạn sẽ thu được kết quả như Hình 8.4.

Như vậy chương trình server đơn giản bạn viết thực sự có thể tiếp nhận truy vấn từ trình duyệt và trả lại kết quả cho trình duyệt.



```
Waiting for connection ...
GET / HTTP/1.1
Waiting for connection ...
GET /favicon.ico HTTP/1.1
Waiting for connection ...

Waiting for connection ...
```

Hình 8.4. Chương trình server cung cấp trang web cho trình duyệt

Trong ví dụ đơn giản trên, chúng ta đã thực hiện những thao tác sau:

(1) Khởi tạo một TCP socket gắn với mọi giao diện mạng trên máy cục bộ và lắng nghe cổng TCP/80

```
TcpListener listener = new TcpListener(IPAddress.Any, 80);
listener.Start(5);
```

(2) Tiếp nhận yêu cầu kết nối TCP từ trình duyệt (hoặc bất kỳ chương trình nào khác)

```
var client = listener.AcceptTcpClient();
```

(3) Đọc dòng request line của truy vấn HTTP sử dụng adapter StreamReader

```
var sr = new StreamReader(client.GetStream());
string request = sr.ReadLine();
```

(4) Tạo một chuỗi phản hồi đơn giản chứa status line và body là một chuỗi HTML, và gửi chuỗi phản hồi này trở lại cho client.

```
var sw = new StreamWriter(client.GetStream());
sw.WriteLine("HTTP/1.1 200 OK\n");
sw.WriteLine("<html><head></head><body><h1>Hello world</h1><body></html>");
```

Như vậy, bạn có thể thấy, chương trình do chúng ta viết ra có thể tương tác với trình duyệt thông qua giao thức HTTP. Bạn cũng có thể thấy, chuỗi HTML cũng được chứa bên trong phản hồi. Bên trong phản hồi này có thể chứa mã JavaScript, CSS, hoặc bất kỳ thứ gì có thể chuyển thành văn bản.

8.2.3. Hệ thống web và giao thức HTTP

a) Hệ thống web

Hệ thống web là một loại ứng dụng mạng bao gồm hai thành phần chính là chương trình chủ web và trình duyệt web. Trong đó:

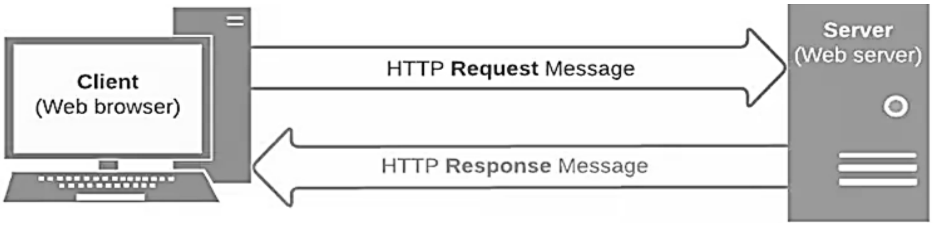
Chương trình chủ web (web server) là một máy tính được thiết lập để chạy các ứng dụng web và cung cấp các tài nguyên như các trang web, tập tin và các ứng dụng trực tuyến khác. Cơ sở dữ liệu là nơi lưu trữ dữ liệu cho các ứng dụng web và cung cấp các dịch vụ truy vấn và lưu trữ dữ liệu.

Trình duyệt web (web browser hay web client) là phần mềm được cài đặt trên máy tính cá nhân hoặc thiết bị di động để truy cập các trang web và ứng dụng trên mạng Internet. Trình duyệt web giúp người dùng truy cập các nội dung trực tuyến, tương tác với các ứng dụng web và sử dụng các dịch vụ trực tuyến khác.

Ở đây chú ý phân biệt giữa hệ thống web và ứng dụng web. Ứng dụng web thường dùng để chỉ loại phần mềm chạy trên máy chủ web kết hợp với chương trình chủ web để sinh ra dữ liệu ở dạng HTML. Ứng dụng web thường viết bằng các ngôn ngữ như PHP, C#, Java, và sử dụng một trong số các framework hỗ trợ cho ngôn ngữ đó.

Trong một hệ thống web, giao thức HTTP được sử dụng để trao đổi dữ liệu giữa chương trình chủ web và trình duyệt web.

HTTP hoạt động dựa trên mô hình client-server, trong đó các trình duyệt web (client) gửi các yêu cầu HTTP đến máy chủ web (server), và server trả về các phản hồi HTTP chứa các tài liệu được yêu cầu.



Hình 8.5. Mô hình hoạt động của hệ thống web

Quá trình hoạt động của HTTP bao gồm các bước sau:

1) Client gửi yêu cầu HTTP đến server: Khi người dùng nhập địa chỉ URL vào trình duyệt, trình duyệt sẽ tạo ra yêu cầu HTTP và gửi nó đến server sử dụng dịch vụ truyền thông của TCP.

2) Server nhận yêu cầu HTTP: Khi server nhận được yêu cầu HTTP từ client, nó sẽ xử lý yêu cầu này để trả về tài liệu được yêu cầu.
3) Server xử lý yêu cầu HTTP: Server sẽ kiểm tra yêu cầu HTTP để đảm bảo tính hợp lệ của yêu cầu, xác thực người dùng (nếu cần thiết) và tìm kiếm tài liệu được yêu cầu.
4) Server trả về phản hồi HTTP: Sau khi server xử lý yêu cầu, nó sẽ trả về phản hồi HTTP chứa tài liệu được yêu cầu. Phản hồi này bao gồm các tiêu đề HTTP, mã trạng thái và nội dung của tài liệu.
5) Client nhận phản hồi HTTP: Sau khi client nhận được phản hồi HTTP, trình duyệt sẽ hiển thị tài liệu được yêu cầu cho người dùng.

b) Cấu trúc gói tin của HTTP

Như ở trên đã trình bày, giao thức HTTP có hai loại gói tin với cấu trúc khác nhau: gói tin truy vấn (HTTP request) và gói tin phản hồi (HTTP response). Mỗi loại gói tin thực chất chỉ là các chuỗi văn bản có định dạng xác định.

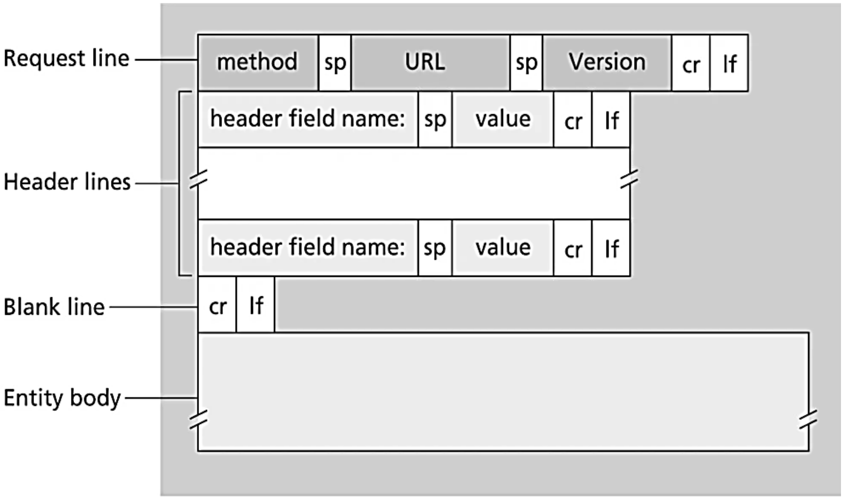
Gói tin truy vấn của HTTP bao gồm các phần sau (Hình 8.6):

Method (phương thức): xác định hành động mà client muốn thực hiện trên tài nguyên của server, ví dụ như GET (để lấy thông tin), POST (để tạo mới tài nguyên), PUT (để cập nhật tài nguyên), DELETE (để xóa tài nguyên), và nhiều phương thức khác.

URL (Uniform Resource Locator): xác định địa chỉ của tài nguyên mà client muốn truy cập trên server.

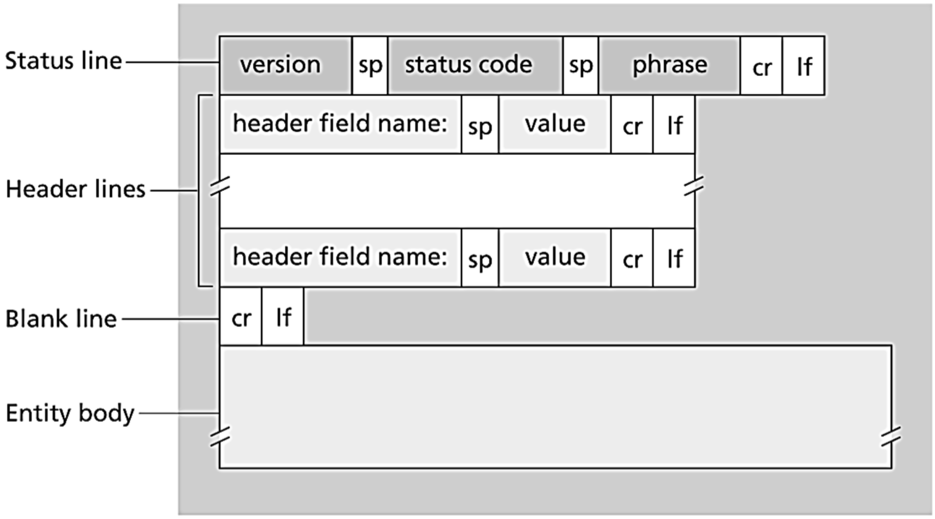
Headers (tiêu đề): chứa các thông tin về yêu cầu của client, ví dụ như kiểu dữ liệu yêu cầu, ngôn ngữ yêu cầu, thông tin xác thực, và các thông tin khác.

Body (nội dung): chứa các thông tin cụ thể mà client muốn gửi đến server, ví dụ như dữ liệu form, file tải lên, v.v.



Hình 8.6. Cấu trúc gói tin truy vấn HTTP

Gói tin phản hồi của HTTP bao gồm các phần sau (Hình 8.7):


Status line
version
sp
status code
sp
phrase
cr
lf
Header lines
header field name:
sp
value
cr
lf
header field name:
sp
value
cr
lf
Blank line
cr
lf
Entity body

Hình 8.7. Cấu trúc gói tin phản hồi HTTP

1) Status code (mã trạng thái): xác định kết quả của yêu cầu của client, ví dụ như thành công (200 OK), tài nguyên không tìm thấy (404 Not Found), lỗi server (500 Internal Server Error), v.v.
2) Headers (tiêu đề): chứa các thông tin về phản hồi của server, ví dụ như kiểu dữ liệu trả về, ngôn ngữ phản hồi, thông tin xác thực, và các thông tin khác.
3) Body (nội dung): chứa các thông tin cụ thể mà server muốn trả về cho client, ví dụ như nội dung HTML, JSON, XML, v.v.

Như vậy, nếu biết cấu trúc chuỗi văn bản này và biết cách lập trình với giao thức TCP, chúng ta hoàn toàn có thể tự xây dựng một chương trình web server đơn giản, hoặc có thể xây dựng một chương trình để phát truy vấn HTTP tới chương trình chủ.

8.2.4. Cấu trúc địa chỉ web

Địa chỉ web, thường gọi là URL (Uniform Resource Locator), là địa chỉ duy nhất dùng để xác định và truy cập một tài nguyên (resource) trên Internet - chẳng hạn như một trang web, một hình ảnh, hoặc một file. Có thể hiểu URL giống như địa chỉ nhà của một tài nguyên trên mạng.

Cấu trúc của một URL tiêu chuẩn có thể chia thành các phần như sau:

```
scheme://username:password@host:port/path?query#fragment

Ví dụ:
https://www.example.com:443/path/to/page.html?search=abc#top
```

Trong đó:

Thành phần	Ý nghĩa
scheme	Giao thức truy cập tài nguyên, ví dụ: http, https, ftp, mailto
host	Tên miền hoặc địa chỉ IP của máy chủ (ví dụ: www.example.com)
port *(tuỳ chọn)*	Cổng dịch vụ (mặc định: 80 cho HTTP, 443 cho HTTPS)
path	Đường dẫn đến tài nguyên cụ thể trên máy chủ
query *(tuỳ chọn)*	Tham số truyền qua URL, thường bắt đầu với dấu ? (ví dụ: ?id=123)
fragment *(tuỳ chọn)*	Đoạn con của trang (ví dụ: #section1)

Hãy xem ví dụ về một URL đầy đủ

```
https://news.example.com:443/articles/view.html?id=567#comments
```

- https: Giao thức bảo mật HTTP.
- news.example.com: Tên miền máy chủ chứa nội dung.
- 443: Cổng sử dụng (443 là mặc định cho HTTPS).
- /articles/view.html: Đường dẫn đến tệp HTML cụ thể.
- ?id=567: Truy vấn lấy bài viết có mã số 567.
- #comments: Trỏ tới phần bình luận của trang.

Một số lưu ý quan trọng:

- URL phân biệt chữ hoa chữ thường ở phần path, nhưng không phân biệt ở phần host.
- Trình duyệt sẽ tự động thêm giao thức (ví dụ http://) nếu người dùng không chỉ rõ.
- Việc mã hoá URL là cần thiết với các ký tự đặc biệt (ví dụ: khoảng trắng → %20).

8.2.5. Truyền dữ liệu qua HTTP

Giao thức HTTP được sử dụng để truyền tải nhiều loại dữ liệu khác nhau giữa client và chương trình chủ web. Các loại dữ liệu thông dụng nhất bao gồm HTML, CSS, JavaScript, file ảnh, dữ liệu dạng JSON, dữ liệu dạng XML.

a) Truyền các loại dữ liệu web truyền thống

HTML (HyperText Markup Language) là loại ngôn ngữ dùng để tạo ra nội dung trên web. Bản thân dữ liệu HTML là văn bản, chứa các đánh dấu (markup), do vậy, có thể dễ dàng nhúng vào phần body của gói tin HTTP response để trả cho trình duyệt.

CSS (Cascading Style Sheets) là ngôn ngữ dùng để mô tả cách trình bày và hiển thị các trang web được viết bằng HTML. File CSS thực chất cũng là một file văn bản, vì vậy nó có thể được nhúng trực tiếp vào phần body của gói tin HTTP response.

JavaScript là loại ngôn ngữ lập trình được sử dụng phổ biến trên trình duyệt web để tạo ra các ứng dụng web động và tương tác trên trang web. File JavaScript có phần mở rộng js cũng là một file văn bản. Phản hồi của HTTP chứa mã nguồn của JavaScript trực tiếp trong phần body để gửi cho trình duyệt.

Thông thường, một trang web được tạo ra từ một file HTML (chứa nội dung). Mỗi file HTML thường tham chiếu tới rất nhiều file CSS (chứa mô tả cách trình bày trang) và JavaScript (tạo tương tác động). Mỗi khi file HTML được tải về, trình duyệt sẽ căn cứ vào các tham chiếu chứa trong HTML để lần lượt tải tất cả các file có liên quan về. Chỉ khi nào tải được đầy đủ các file, trang web mới thực sự hoạt động.

b) Truyền các loại dữ liệu đặc biệt

Với các ứng dụng đơn trang, ứng dụng desktop, hoặc ứng dụng mobile, giao thức HTTP không được sử dụng để truyền tải các file. Do HTTP sử dụng văn bản cho cấu trúc gói tin, bất kỳ văn bản nào cũng có thể gửi qua giao thức này. Có hai định dạng văn bản đặc biệt được sử dụng phổ biến: JSON và XML. Chương trình server gửi các chuỗi ký tự JSON hoặc XML bên trong phần body.

JSON là một loại định dạng dữ liệu được viết theo cú pháp tạo đối tượng của ngôn ngữ JavaScript. Thực chất JSON cũng chỉ là một chuỗi ký tự được chương trình server sinh ra theo yêu cầu từ client.

XML là một loại ngôn ngữ định dạng dữ liệu khác cũng ở dạng văn bản. Chuỗi XML được chương trình server nhúng vào phần body của gói tin response.

Hai loại văn bản đặc biệt này được sử dụng để định dạng dữ liệu trao đổi giữa ứng dụng Web API và ứng dụng khai thác nó, như ứng dụng mobile, ứng dụng desktop, ứng dụng web đơn trang. Cách xây dựng ứng dụng dạng Web API sẽ được xem xét trong một bài học riêng.

c) Truyền dữ liệu nhị phân

Dữ liệu nhị phân là những loại dữ liệu khác mà không thể hiện trực tiếp thành văn bản. Dữ liệu nhị phân thường dùng để lưu trữ các tập tin ảnh, âm thanh, video, phần mềm, v.v..

Đối với **dữ liệu nhị phân**, cần phải có cách thức riêng để chuyển đổi nó thành văn bản trước khi có thể nhúng vào thân của truy vấn / phản hồi HTTP. Để chuyển đổi file nhị phân thành text để truyền qua giao thức HTTP, bạn có thể sử dụng thuật toán mã hóa nhị phân sang Base64.

Base64 là một thuật toán mã hóa dữ liệu nhị phân thành dạng văn bản có thể đọc được bằng cách sử dụng bảng ký tự định sẵn. Trong quá trình chuyển đổi, các ký tự không hợp lệ hoặc không thể đọc được sẽ được thay thế bằng các ký tự an toàn.

Để chuyển đổi file nhị phân sang Base64, bạn có thể sử dụng các thư viện mã hóa Base64 có sẵn trong các ngôn ngữ lập trình. Sau khi chuyển đổi file nhị phân sang chuỗi Base64, bạn có thể truyền dữ liệu qua giao thức HTTP như với bất kỳ dữ liệu văn bản nào.

d) Cơ chế truyền dữ liệu trong HTTP

Do sử dụng dịch vụ truyền TCP, giao thức HTTP có hai cơ chế truyền dữ liệu: persistent (keep-alive) và non-persistent.

Trong cơ chế **non-persistent** mỗi yêu cầu HTTP được gửi qua một kết nối mới giữa máy khách và máy chủ và kết nối sẽ bị ngắt sau khi phản hồi HTTP được trả về cho máy khách. Cơ chế này giúp tối ưu hóa sử dụng tài nguyên mạng và giảm thiểu khối lượng thông tin được gửi qua mạng. Tuy nhiên, nó cũng tăng thời gian phản hồi do mỗi yêu cầu phải thiết lập một kết nối mới.

Trong cơ chế **persistent** (còn gọi là "keep-alive") kết nối giữa máy khách (client) và máy chủ (server) được duy trì cho nhiều vòng phát truy vấn và nhận phản hồi. Thay vì mỗi yêu cầu từ máy khách tạo ra một kết nối mới đến máy chủ và ngắt kết nối sau khi phản hồi được trả về, kết nối keep-alive cho phép nhiều yêu cầu được gửi trên cùng một kết nối giữa máy khách và máy chủ. Điều này giảm thiểu chi phí kết nối và giảm độ trễ (latency) trong việc truyền tải dữ liệu giữa máy khách và máy chủ.

Các trình duyệt web hiện đại thường tự động sử dụng kết nối keep-alive khi gửi yêu cầu HTTP đến máy chủ. Tuy nhiên, một số máy chủ và ứng dụng web cũ hơn có thể không hỗ trợ kết nối keep-alive hoặc có thể tắt tính năng này để giảm tải cho máy chủ.

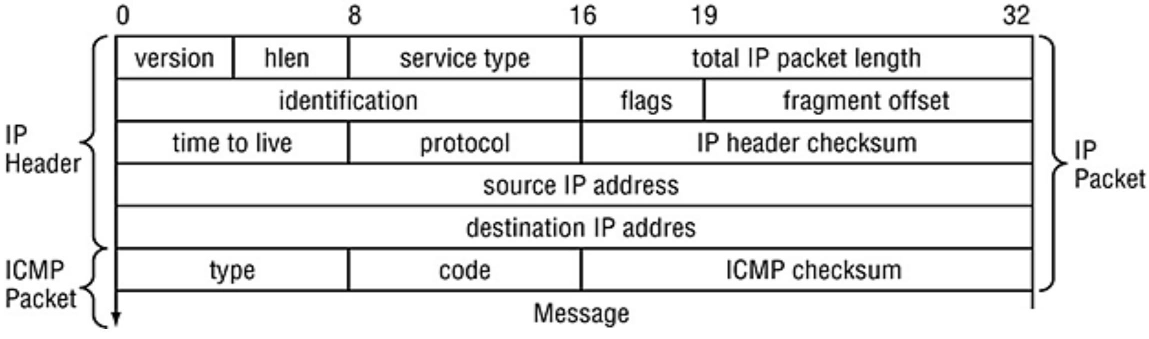
Việc hiểu hai cơ chế truyền dữ liệu này có vai trò quan trọng trong việc tối ưu hóa các ứng dụng web. Nếu máy chủ sử dụng cơ chế non-persistent, ứng dụng web thường cố gắng tạo ra ít file nhất có thể, bằng cách gộp nhiều file css/js nhỏ thành một file lớn để gửi đi trong 1 phản hồi. Nếu máy chủ sử dụng cơ chế keep-alive, việc gộp file này không cần thiết nữa.

8.3. Lập trình ứng dụng Ping với ICMP

8.3.1. Giao thức ICMP

Để minh họa cách làm việc với socket thô, chúng ta sẽ cùng xây dựng một chương trình mô phỏng hoạt động của ứng dụng ping sử dụng giao thức ICMP. Trước khi lập trình, chúng ta cần hiểu cách hoạt động của giao thức này.

ICMP - định nghĩa trong RFC 792 - là một giao thức được thực thi trên nhiều loại thiết bị mạng và thiết bị đầu cuối giúp các thiết bị này trao đổi thông tin. Đây là giao thức được sử dụng trong các ứng dụng hỗ trợ quản lý mạng như ping hay trace route.


0
8
16
19
32
version
hlen
service type
total IP packet length
identification
flags
fragment offset
time to live
protocol
IP header checksum
source IP address
destination IP addres
type
code
ICMP checksum
Message
IP Header
ICMP Packet
IP Packet

Hình 8.8. Gói tin của ICMP đặt trong gói tin IP

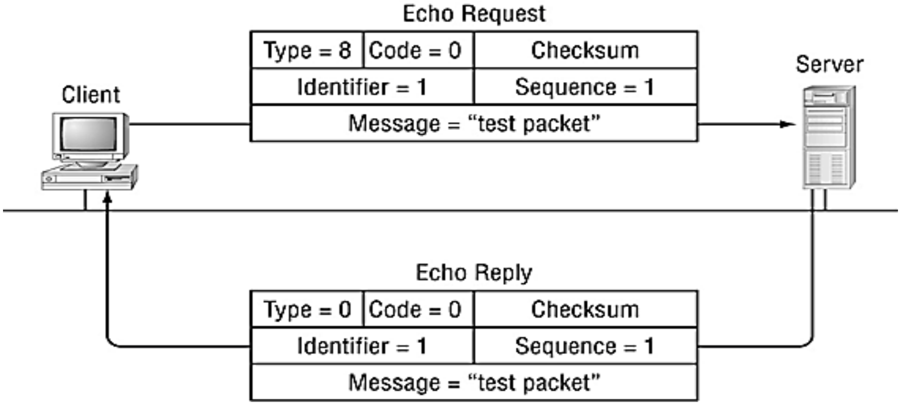
ICMP là một giao thức khá đặc biệt. Bản thân nó khai thác dịch vụ truyền thông host-to-host của giao thức IP (tương tự như TCP và UDP), đồng thời toàn bộ gói tin ICMP được đặt trong phần dữ liệu của gói tin IP. Tuy nhiên ICMP lại không phải là giao thức thuộc tầng giao vận, tức là nó không cùng cấp độ với TCP và UDP.

ICMP không được sử dụng để mang dữ liệu người dùng. Thay vào đó, nó được sử dụng như một công cụ để hỗ trợ trao đổi thông điệp giữa các thiết bị thực hiện truyền thông với giao thức IP.

Gói tin ICMP có cấu trúc khá đơn giản:

- Trường type (1 byte): xác định loại thông điệp. Giá trị của trường type cũng được gọi là Type Code. Mỗi Type Code có ý nghĩa riêng. Ví dụ: Type Code 0 - echo reply (gói tin phản hồi của thông điệp echo); Type Code 8 - Echo Request (gói tin yêu cầu của thông điệp echo). Thông điệp echo được sử dụng cho ứng dụng ping. Chúng ta sẽ quay lại với thông điệp echo sau.
- Trường Code (1 byte): xác định thông tin chi tiết và làm rõ hơn ý nghĩa mô tả trong trường type. Ví dụ, Code 0 - Network Unreachable, Code 1 - Host Unreachable.
- Trường Checksum (2 byte) - dùng để kiểm soát lỗi. Trường này được tính theo một thuật toán riêng.
- Trường Message có độ dài biến đổi và có thể chứa nhiều loại thông tin khác nhau. Hai loại thông tin thường gặp trong trường Mesage là số định danh của gói tin (Identifier) và số thứ tự (Sequence Number).

Ping (Hình 6.3) là một trong những ứng dụng phổ biến của ICMP. Ứng dụng này cho phép kiểm tra xem có liên kết mạng giữa máy đang dùng với một thiết bị mạng khác (như router, thiết bị đầu cuối, gateway) hay không.



Hình 8.9. Sử dụng giao thức ICMP trong chương trình ping

Ứng dụng Ping sử dụng mô hình truyền thông kiểu request / response. Do đó, ứng dụng Ping sử dụng hai loại gói tin ICMP:

- Gói tin Echo Request do thiết bị hiện tại phát tới thiết bị đích: Gói tin Echo Request có Type = 8 và chứa một thông điệp bất kỳ.
- Gói tin Echo Reply cho thiết bị đích phát trở lại cho thiết bị nguồn: Gói tin Echo Reply có Type = 0 và thông điệp lặp lại từ Echo Request. Khi thiết bị đích nhận được gói tin Echo Request, nó sẽ gửi lại gói tin Echo Reply cho thiết bị nguồn.

Nếu nhận được gói tin Echo Reply, nghĩa là máy hiện tại và thiết bị đích có kết nối mạng với nhau. Nếu không nhận được gói Echo Reply, liên kết mạng đang không hoạt động.

8.3.2. Thực hành: Xây dựng ứng dụng ping

Chúng ta cùng xây dựng chương trình ping dựa trên những kiến thức vừa học. Chương trình này sử dụng socket thô để gửi gói tin echo request của ICMP và nhận lại gói tin echo reply.

Bước 1: Tạo một dự án mới thuộc kiểu ConsoleApp và đặt tên là Ping.

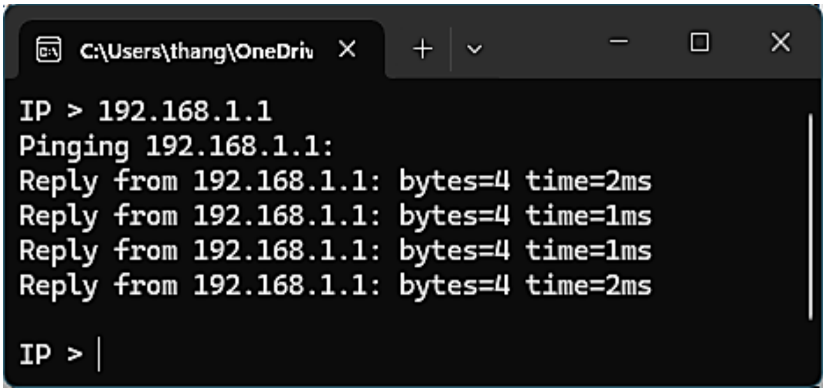
*Bước 2:*Viết code cho file Program.cs như sau:

Program.cs

```
using System.Net;
using System.Net.Sockets;
using System.Diagnostics;
using System.Text;

while (true) {
   Console.Write("IP > ");
   if (!IPAddress.TryParse(Console.ReadLine(), out var ip)) continue;
   var sock = new Socket(AddressFamily.InterNetwork, SocketType.Raw,
ProtocolType.Icmp);
   sock.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.ReceiveTimeout,
3000);
   var msg = Encoding.ASCII.GetBytes("Ping");
   var id = (ushort)Environment.ProcessId;
   var buf = new byte[1024];
   var ep = new IPEndPoint(ip, 0);
   var recv = new IPEndPoint(IPAddress.Any, 0) as EndPoint;

   Console.WriteLine($"Pinging {ip}:");

   for (int i = 0; i < 4; i++) {
      var pkt = new byte[8 + msg.Length];
      pkt[0] = 8;
      pkt[4] = (byte)(id >> 8); pkt[5] = (byte)id;
      pkt[6] = 0; pkt[7] = (byte)i;
      Array.Copy(msg, 0, pkt, 8, msg.Length);
```

```
        var sum = 0;
        for (int j = 0; j < pkt.Length; j += 2)
            sum += (j + 1 < pkt.Length) ? (pkt[j] << 8) + pkt[j + 1] : pkt[j] << 8;
        sum = (sum >> 16) + (sum & 0xFFFF);
        sum += sum >> 16;
        var cksum = (ushort)~sum;
        pkt[2] = (byte)(cksum >> 8); pkt[3] = (byte)cksum;

        var sw = Stopwatch.StartNew();
        sock.SendTo(pkt, ep);
        try {
            int len = sock.ReceiveFrom(buf, ref recv);
            sw.Stop();
            Console.WriteLine($"Reply from {ip}: bytes={len - 28}
time={sw.ElapsedMilliseconds}ms");
        } catch { Console.WriteLine("Timeout"); }

        Thread.Sleep(1000);
    }

    sock.Close();
    Console.WriteLine();
}
```

Bước 3: Chạy thử nghiệm chương trình từ giao diện dòng lệnh. Nhập một địa chỉ IP và ấn enter, chúng ta sẽ thu được kết quả như Hình 8.10. Kết quả của chương trình ping này tương tự như cách ứng dụng ping của hệ thống hoạt động.



```
IP > 192.168.1.1
Pinging 192.168.1.1:
Reply from 192.168.1.1: bytes=4 time=2ms
Reply from 192.168.1.1: bytes=4 time=1ms
Reply from 192.168.1.1: bytes=4 time=1ms
Reply from 192.168.1.1: bytes=4 time=2ms

IP > |
```

Hình 8.10. Kết quả hoạt động của chương trình ping vừa xây dựng

Mặc định chương trình sẽ gửi đi 4 gói tin ICMP echo request. Mỗi gói tin này chứa PID của tiến trình đang hoạt động, một số thứ tự, và chuỗi thông điệp `Ping`. Cứ mỗi 1,5 giây chương trình sẽ phát đi một gói tin ICMP echo request như vậy.

Ứng với mỗi gói request phát đi, chương trình sẽ chờ nhận gói tin ICMP echo response trong 3 giây. Nếu quá 3 giây mà chưa nhận được phản hồi, chương trình coi đây là lỗi. Thời gian từ lúc phát gói tin đi đến lúc nhận được gói tin phản hồi về được tính bằng đơn vị ticks và quy đổi ra mili giây. Nếu ping trong mạng LAN hoặc địa chỉ loopback, thời gian này thường rất nhỏ.

BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

1) Mở rộng chương trình web server đã xây dựng ở bài thực hành 8.2.2 cho phép cung cấp các file html, css và js đã xây dựng sẵn theo yêu cầu của trình duyệt.
2) Mở rộng chương trình web server cho phép cung cấp file ảnh (bmp, jpg, png) cho trình duyệt. Gợi ý, sử dụng thư viện Base64 để chuyển đổi file nhị phân về chuỗi ký tự.
3) Mở rộng chương trình ping đã xây dựng ở bài thực hành 8.3.2 cho phép xử lý những tình huống ngoại lệ (như gói tin request không tới đích, mạng bị lỗi).

Câu hỏi lý thuyết

1) Tại sao khi xây dựng ứng dụng mạng, lập trình viên gần như bắt buộc phải thiết kế hoặc sử dụng một giao thức tầng ứng dụng? Việc này ảnh hưởng thế nào đến khả năng tương tác của phần mềm?
2) Trong lập trình socket với TCP, điều gì sẽ xảy ra nếu client gửi dữ liệu nhưng không gọi hàm Flush() sau Write()? Tình huống này phản ánh nguyên lý gì về hoạt động của luồng dữ liệu?
3) Phân tích sự khác biệt giữa hai mô hình: “ứng dụng sử dụng HTTP” và “ứng dụng tự định nghĩa giao thức tầng ứng dụng sử dụng TCP”. Trong trường hợp nào nên chọn phương án thứ hai?
4) Tại sao mô hình publish/subscribe lại phù hợp với các ứng dụng IoT hoặc giám sát thiết bị phân tán? Mô hình này khắc phục hạn chế nào của mô hình client/server truyền thống?
5) Lập trình viên có thể sử dụng TcpClient trong .NET để kết nối đến máy chủ web. Nhưng tại sao việc xây dựng request theo đúng cú pháp HTTP lại là trách nhiệm của người lập trình? Điều này thể hiện gì về vai trò của giao thức tầng ứng dụng?
6) So sánh giữa Fat Client và Thin Client. Trong môi trường mạng có tốc độ chậm, bạn sẽ ưu tiên thiết kế theo mô hình nào? Giải thích.

7) Nếu sử dụng mô hình pipeline để thiết kế ứng dụng xử lý ảnh theo chuỗi (lọc - nén - gửi), mỗi bước xử lý nên là pusher hay puller? Tại sao việc tách rời theo pipeline lại giúp dễ mở rộng hệ thống?
8) Khi xây dựng một ứng dụng HTTP client đơn giản, tại sao chỉ cần thao tác với chuỗi văn bản đúng định dạng đã có thể tương tác với server thực tế như IIS hoặc Apache? Điều này nói lên điều gì về thiết kế của HTTP?
9) Giả sử bạn thay mô hình client/server bằng mô hình peer-to-peer cho một ứng dụng chia sẻ file. Những thay đổi nào về mặt kiến trúc lập trình và quản lý kết nối sẽ xuất hiện?
10) Khi sử dụng socket thô để lập trình ứng dụng Ping, tại sao phải thiết lập đúng checksum và định dạng gói ICMP thủ công? Trường hợp này có gì khác với khi sử dụng thư viện cao cấp như System.Net.Ping trong .NET?

BÀI 9. CÔNG NGHỆ VÀ DỊCH VỤ MẠNG HIỆN ĐẠI

Mục tiêu của bài học

Bài học cung cấp cho sinh viên:

- Kiến thức cơ bản về công nghệ mạng LAN, bao gồm các công nghệ có dây và không dây, đặc điểm tính chất của từng công nghệ.
- Kiến thức có bản về công nghệ mạng WAN, khái niệm công nghệ mạng WAN (Internet) gia đình. Hiểu về công nghệ FTTH và giao thức PPPoE trong WAN (Internet) gia đình.
- Kiến thức về các công nghệ mạng tương lai.
- Kỹ năng thực hành cấu hình PPPoE trên thiết bị.

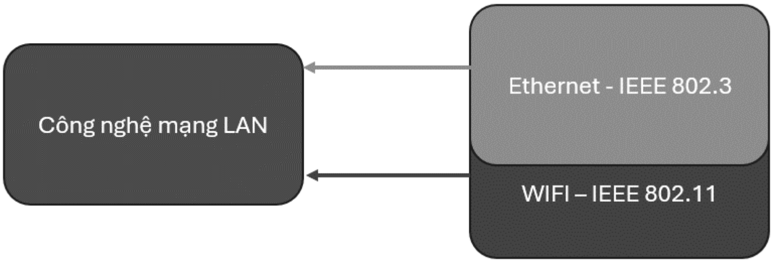
Sau khi hoàn thành bài học, sinh viên có thể:

- Trình bày được các đặc điểm tính chất của các công nghệ LAN không dây và có dây.
- Trình bày được khái niệm công nghệ mạng WAN, công nghệ mạng WAN (Internet) gia đình.
- Hiểu về công nghệ FTTH và các thành phần cơ bản trong một mạng sử dụng FTTH.
- Thực hiện được việc thiết lập một kết nối WAN (Internet) gia đình sử dụng công nghệ FTTH thông qua cấu hình PPPoE.

9.1. Công nghệ mạng

9.1.1. Công nghệ mạng LAN

Mạng LAN (Local Area Network) là một loại mạng máy tính được thiết kế để kết nối các thiết bị trong một khu vực địa lý nhỏ như văn phòng, trường học hoặc nhà ở. Để hiện thực hóa mạng LAN trong thực tế, người ta cần một công nghệ truyền dẫn dữ liệu hiệu quả - và đó chính là vai trò của Ethernet.



Hình 9.1. Mối liên hệ giữa công nghệ mạng LAN và IEEE 802.3, IEEE 802.11

Ethernet là một công nghệ mạng có dây, được chuẩn hóa theo IEEE 802.3. Ngoài Ethernet, mạng LAN có thể được triển khai bằng công nghệ không dây như Wi-Fi (IEEE 802.11). Tìm hiểu công nghệ mạng LAN chính là đi tìm hiểu công nghệ IEEE 802.3 và IEEE 802.11.

a) Các chuẩn IEEE 802

Ethernet là một công nghệ mạng phổ biến được sử dụng rộng rãi trong việc kết nối các thiết bị trong mạng máy tính. Nó định nghĩa các giao thức và tiêu chuẩn cho việc truyền dữ liệu giữa các thiết bị trong mạng cáp. Các chuẩn Ethernet được quy định và duy trì bởi Hiệp hội Kỹ sư Điện và Điện tử (Institute of Electrical and Electronics Engineers - IEEE).

Dưới đây là một số chuẩn IEEE 802 phổ biến trong họ chuẩn Ethernet:

- IEEE 802.3 (Ethernet): Chuẩn Ethernet gốc, được phát triển từ những năm 1980. Nó xác định các tiêu chuẩn về truyền dẫn dữ liệu qua cáp mạng, bao gồm cáp đồng trục, cáp xoắn đôi và cáp quang.
- IEEE 802.3u (Fast Ethernet): Chuẩn Ethernet tốc độ cao, cung cấp tốc độ truyền dẫn dữ liệu 100 megabit/giây (Mbps). Nó sử dụng cáp xoắn đôi thông thường.
- IEEE 802.3z (Gigabit Ethernet): Chuẩn Ethernet siêu tốc, cung cấp tốc độ truyền dẫn dữ liệu 1 gigabit/giây (Gbps). Nó sử dụng cáp quang đơn mạch hoặc cáp xoắn đôi chéo.
- IEEE 802.3ae (10 Gigabit Ethernet): Chuẩn Ethernet 10 gigabit/giây (10 Gbps), thường được sử dụng trong các môi trường mạng doanh nghiệp và trung tâm dữ liệu.
- IEEE 802.3ba (40 Gigabit Ethernet và 100 Gigabit Ethernet): Chuẩn Ethernet 40 Gbps và 100 Gbps, thường được sử dụng trong các trung tâm dữ liệu và mạng cốt lõi.
- Ngoài ra, còn có các chuẩn Ethernet khác như IEEE 802.3af (Power over Ethernet - PoE) cho phép truyền dữ liệu và nguồn điện qua cùng một dây cáp, và IEEE 802.3at (PoE+) cho phép cung cấp nguồn điện mạnh hơn qua cáp Ethernet.

Các chuẩn Ethernet trong họ chuẩn IEEE 802 đóng vai trò quan trọng trong việc xác định tốc độ và phạm vi truyền dẫn dữ liệu trong mạng Ethernet, đáp ứng nhu cầu truyền thông ngày càng cao của các ứng dụng mạng hiện đại.

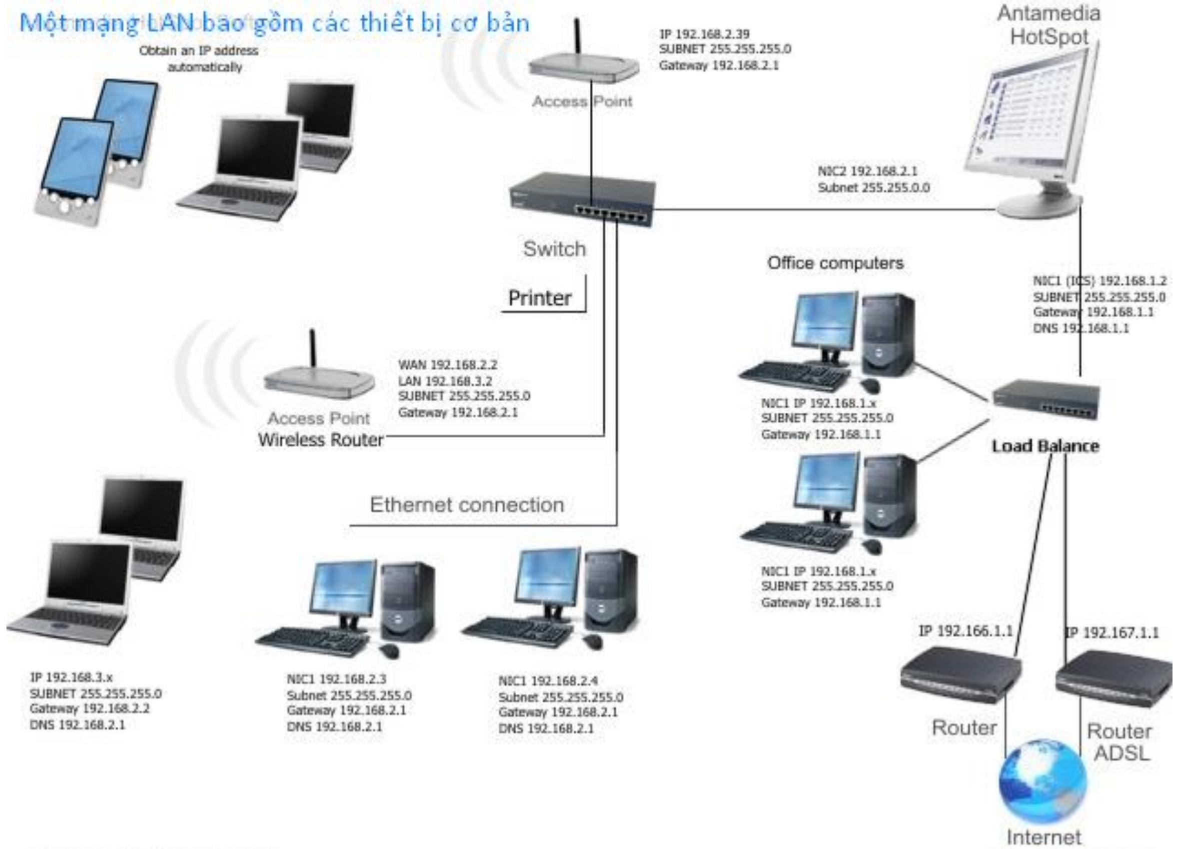
b) Các chuẩn IEEE 802.11

Wireless là một công nghệ mạng cho phép truyền dẫn dữ liệu giữa các thiết bị mà không cần sử dụng cáp vật lý. Công nghệ Wireless được định nghĩa và quản lý bởi Hiệp hội Kỹ sư Điện và Điện tử (Institute of Electrical and Electronics Engineers - IEEE). Trong lĩnh vực Wireless, chuẩn IEEE 802.11 là chuẩn phổ biến được sử dụng cho mạng Wi-Fi.

Dưới đây là một số chuẩn IEEE 802.11 phổ biến trong họ chuẩn Wi-Fi:

- IEEE 802.11b: Chuẩn Wi-Fi đầu tiên được phát triển, hoạt động trên tần số 2.4 GHz và đạt tốc độ truyền dữ liệu lên đến 11 megabit/giây (Mbps).
- IEEE 802.11a: Chuẩn Wi-Fi hoạt động trên tần số 5GHz và đạt tốc độ truyền dữ liệu lên đến 54 Mbps.
- IEEE 802.11g: Chuẩn Wi-Fi hoạt động trên tần số 2.4 GHz và đạt tốc độ truyền dữ liệu lên đến 54 Mbps. Nó tương thích ngược với chuẩn 802.11b.

- IEEE 802.11n: Chuẩn Wi-Fi tốc độ cao, hoạt động trên cả tần số 2.4 GHz và 5 GHz, và đạt tốc độ truyền dữ liệu lên đến 600 Mbps. Nó hỗ trợ công nghệ MIMO (Multiple-Input Multiple-Output) để tăng hiệu suất và độ bao phủ sóng.
- IEEE 802.11ac: Chuẩn Wi-Fi siêu tốc, hoạt động trên tần số 5GHz và đạt tốc độ truyền dữ liệu lên đến vài gigabit/giây (Gbps). Chuẩn này sử dụng công nghệ MIMO và có khả năng hỗ trợ nhiều người dùng và ứng dụng đòi hỏi băng thông cao.
- IEEE 802.11ax (Wi-Fi 6): Chuẩn Wifi mới nhất, hoạt động trên cả tần số 2.4 GHz và 5 GHz. Nó cung cấp tốc độ truyền dẫn dữ liệu nhanh hơn, khả năng xử lý tốt hơn đối với nhiều thiết bị kết nối cùng một lúc và hiệu suất mạng tốt hơn trong môi trường đông người dùng.


Một mạng LAN bao gồm các thiết bị cơ bản
Obtain an IP address automatically
Access Point
IP 192.168.2.39
SUBNET 255.255.255.0
Gateway 192.168.2.1
Antamedia HotSpot
NIC2 192.168.2.1
Subnet 255.255.0.0
Switch
Printer
Office computers
NIC1 (ICS) 192.168.1.2
SUBNET 255.255.255.0
Gateway 192.168.1.1
DNS 192.168.1.1
WAN 192.168.2.2
LAN 192.168.3.2
SUBNET 255.255.255.0
Gateway 192.168.2.1
Access Point
Wireless Router
NIC1 IP 192.168.1.x
SUBNET 255.255.255.0
Gateway 192.168.1.1
Load Balance
Ethernet connection
NIC1 IP 192.168.1.x
SUBNET 255.255.255.0
Gateway 192.168.1.1
IP 192.166.1.1
IP 192.167.1.1
IP 192.168.3.x
SUBNET 255.255.255.0
Gateway 192.168.2.2
DNS 192.168.2.1
NIC1 192.168.2.3
Subnet 255.255.255.0
Gateway 192.168.2.1
DNS 192.168.2.1
NIC1 192.168.2.4
Subnet 255.255.255.0
Gateway 192.168.2.1
DNS 192.168.2.1
Router
Router ADSL
Internet

Hình 9.2. Một mạng LAN sử dụng cả công nghệ có dây và không dây

Các chuẩn IEEE 802.11 đóng vai trò quan trọng trong việc xác định tốc độ, phạm vi và khả năng tương thích của mạng Wi-Fi, đáp ứng nhu cầu truyền thông không dây ngày càng tăng trong các môi trường cá nhân, công nghiệp và công cộng.

9.1.2. Công nghệ mạng WAN

WAN là mạng kết nối các mạng LAN hoặc hệ thống mạng nhỏ lại với nhau qua khoảng cách địa lý xa. Ví dụ: mạng Internet, mạng của ngân hàng kết nối các chi nhánh toàn quốc.

Có nhiều các công nghệ WAN: FTTH, Leased Line, MPLS, Frame Relay, ATM, xDSL, LTE/5G, Satellite.

WAN (Wide Area Network) trong phạm vi gia đình thường được hiểu là kết nối Internet mà hộ gia đình sử dụng để truy cập các dịch vụ mạng toàn cầu như web, email, mạng xã hội, streaming, học tập, làm việc từ xa... Trong trường hợp này, Internet chính là một mạng WAN lớn nhất, và mỗi hộ gia đình kết nối vào nó thông qua một nhà cung cấp dịch vụ Internet (ISP - Internet Service Provider).

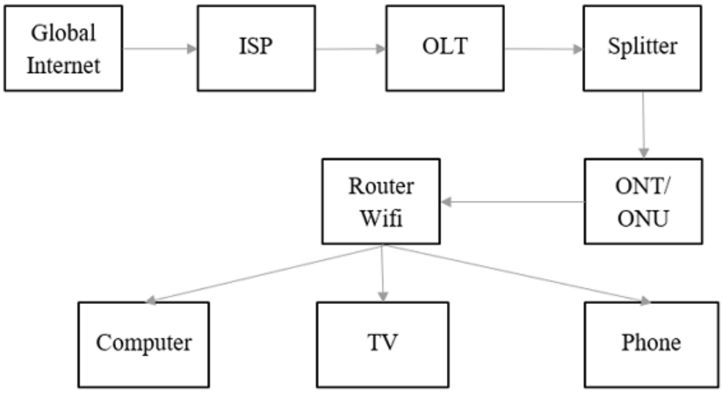
Mạng WAN (Internet) gia đình bao gồm thiết bị đầu cuối (điện thoại, máy tính, TV thông minh…) kết nối với router Wi-Fi hoặc modem, sau đó router sẽ kết nối với ISP thông qua các công nghệ truyền dẫn như FTTH (Fiber To The Home), ADSL, hoặc mạng 4G/5G. Giao thức thường được sử dụng để xác thực và thiết lập kết nối là PPPoE (Point-to-Point Protocol over Ethernet).

Nói cách khác, trong môi trường gia đình, WAN là cổng nối giữa mạng nội bộ (LAN) trong nhà với mạng toàn cầu (Internet). Các thiết bị trong nhà thuộc LAN, còn liên kết ra ngoài (qua modem/router) để truy cập Internet là WAN. Việc quản lý kết nối WAN bao gồm thiết lập thông số kết nối, bảo mật, và tối ưu hiệu suất sử dụng Internet cho các thành viên trong gia đình.

Mặc dù có nhiều công nghệ truyền dẫn và giao thức khác khi nói về Công nghệ mạng WAN, công nghệ FTTH và giao thức PPPoE sẽ được tập trung giới thiệu bởi vì hiện nay FTTH và PPPoE gần gũi với người dùng mạng phổ thông, chúng gắn liền với mạng WAN (internet) gia đình.

a) Công hệ Fiber to Home (FTTH) và mạng WAN (Internet) gia đình

FTTH (Fiber To The Home) là một công nghệ truyền dẫn dữ liệu sử dụng cáp quang để kết nối trực tiếp từ nhà cung cấp dịch vụ đến tận nhà người dùng cuối. FTTH sử dụng sợi quang học - một loại cáp có khả năng truyền tín hiệu ánh sáng với tốc độ rất cao và độ suy hao cực thấp.



Hình 9.3. Sơ đồ mô hình kết nối mạng WAN sử dụng công nghệ FTTH

Công nghệ FTTH có các ưu điểm sau:

- Tốc độ cao: Hỗ trợ tốc độ lên đến hàng Gbps, phù hợp với các ứng dụng như xem video 4K/8K, chơi game online, hội nghị truyền hình.

- Độ ổn định cao: Ít bị nhiễu bởi môi trường điện từ, thời tiết.

- Băng thông rộng: Đáp ứng nhu cầu sử dụng đồng thời nhiều thiết bị trong gia đình hoặc doanh nghiệp.

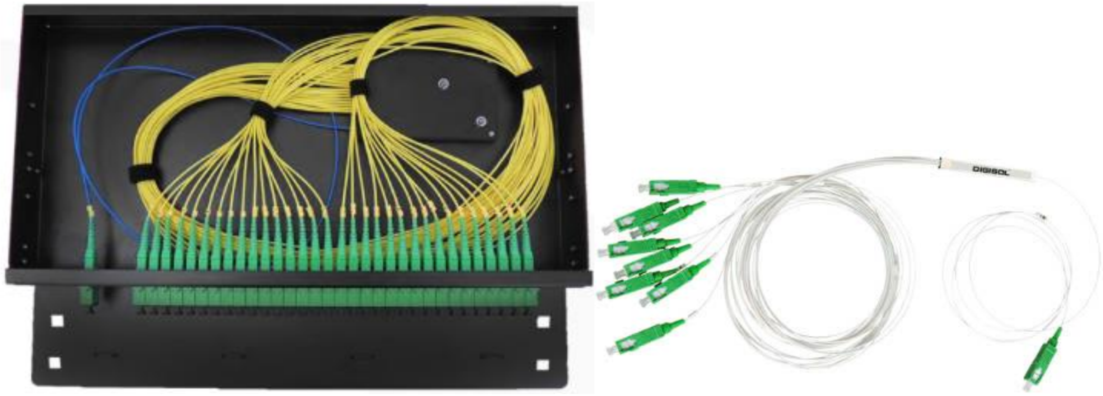
- Tương lai mở rộng: Dễ dàng nâng cấp mà không cần thay đổi hạ tầng vật lý.

Các thành phần trong mô hình kết nối WAN (Internet) gia đình sử dụng công nghệ FTTH bao gồm:

- Global Internet: Mạng Internet toàn cầu mà người dùng cuối muốn truy cập.
- ISP (Nhà cung cấp dịch vụ Internet): Cầu nối giữa người dùng và Internet toàn cầu.
- OLT (Optical Line Terminal): Thiết bị tại trung tâm ISP, quản lý tín hiệu quang.
- Splitter (Bộ chia quang): Chia tín hiệu từ OLT đến nhiều hộ gia đình.
- ONT/ONU (Optical Network Terminal/Unit): Thiết bị tại nhà khách hàng, chuyển đổi tín hiệu quang thành tín hiệu điện.

(a) OLT

(b) ONT/ONU



(c) Splitter

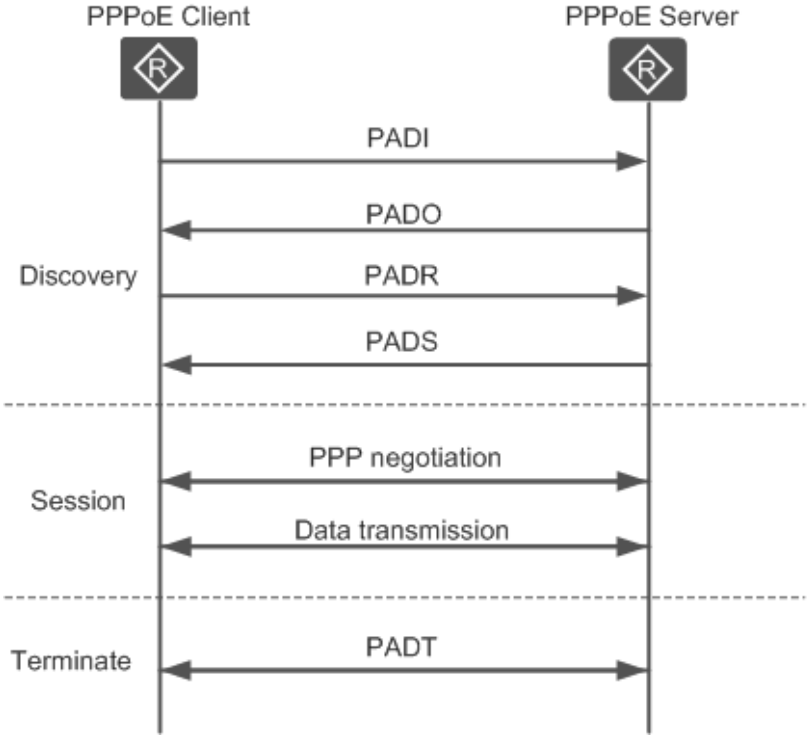
Hình ảnh một số thiết bị sử dụng trong FTTH

b) Giao thức PPPoE trong mạng WAN (Internet gia đình)

PPPoE là một giao thức mạng kết hợp giữa PPP (Point-to-Point Protocol) và Ethernet, cho phép truyền dữ liệu PPP qua mạng Ethernet. Giao thức này thường được sử dụng trong các kết nối Internet băng thông rộng, đặc biệt là trong mạng FTTH (Fiber To The Home), để xác thực người dùng và quản lý phiên kết nối.

Trong quá trình kết nối, PPPoE thực hiện hai giai đoạn chính: Discovery và Session. Ở giai đoạn Discovery, thiết bị người dùng (thường là router hoặc modem) gửi các gói tin PADI để tìm kiếm máy chủ PPPoE của nhà cung cấp dịch vụ (ISP). Sau khi nhận được phản hồi (PADO), thiết bị gửi yêu cầu kết nối (PADR) và nhận xác nhận (PADS) từ ISP, thiết lập một phiên PPP với một Session ID riêng biệt.

Sau khi phiên được thiết lập, giai đoạn Session bắt đầu. Tại đây, người dùng sẽ nhập tên đăng nhập và mật khẩu (do ISP cung cấp) để xác thực. Thông tin này được gửi đến máy chủ xác thực của ISP (thường sử dụng giao thức RADIUS). Nếu thông tin hợp lệ, ISP sẽ cấp phát địa chỉ IP động cho thiết bị người dùng, cho phép truy cập Internet. Quá trình này giúp ISP kiểm soát được ai đang sử dụng dịch vụ, từ đó dễ dàng quản lý băng thông, thời gian sử dụng và bảo mật.



Hình 9.4. Quá trình kết nối và hoạt động của PPPoE

Nhờ cơ chế xác thực và cấp phát IP này, PPPoE trở thành một công cụ quan trọng trong việc quản lý người dùng cuối, đặc biệt trong các hệ thống có số lượng lớn thuê bao như mạng

FTTH dân dụng. Đây cũng là lý do tại sao PPPoE vẫn được sử dụng rộng rãi tại Việt Nam và nhiều quốc gia khác.

Sau khi một Session PPPoE được thiết lập, máy khách PPPoE hoặc máy chủ PPPoE có thể gửi một gói tin PADT dưới dạng unicast để kết thúc (terminate) phiên PPPoE bất kỳ lúc nào. Sau khi truyền hoặc nhận gói tin PADT, máy chủ và máy khách PPPoE không được phép sử dụng phiên này để tiếp tục truyền tin.

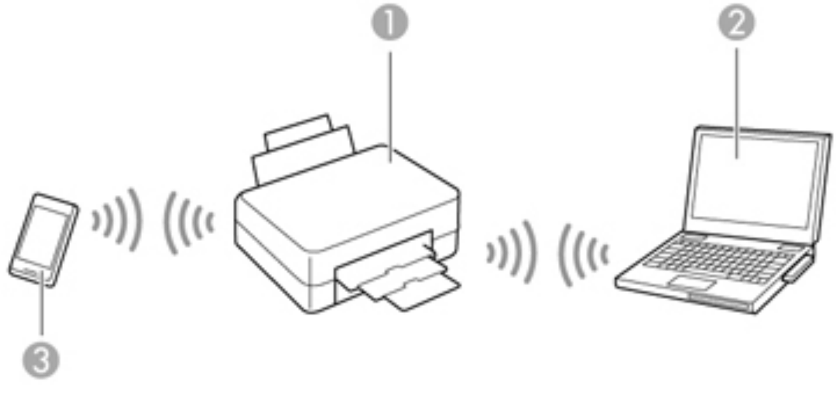
9.1.3. Công nghệ mạng tương lai

a) Wifi Direct

Wi-Fi Direct là một công nghệ kết nối không dây cho phép các thiết bị giao tiếp trực tiếp với nhau mà không cần thông qua điểm truy cập (Access Point) truyền thống. Wi-Fi Direct đóng vai trò quan trọng trong việc đơn giản hóa kết nối giữa các thiết bị di động, máy in, máy ảnh kỹ thuật số, và các thiết bị IoT. Công nghệ này sử dụng chuẩn IEEE 802.11, cho phép thiết lập mạng ngang hàng (peer-to-peer) với tốc độ cao và độ trễ thấp. Đặc biệt, Wi-Fi Direct không chỉ hỗ trợ truyền dữ liệu nhanh chóng mà còn đảm bảo tính bảo mật thông qua các cơ chế mã hóa và xác thực.

Dưới đây là một ví dụ thực tế về việc sử dụng Wi-Fi Direct sử dung trong in tài liệu từ điện thoại (3) sang máy in không dây (1).

Giả sử người dùng có một chiếc điện thoại thông minh và một máy in hỗ trợ Wi-Fi Direct. Người dùng muốn in một tài liệu PDF mà không có sẵn mạng Wi-Fi trong nhà.



Hình 9.5. In ấn hỗ trợ công nghệ Wifi Direct

- Người dùng bật Wi-Fi Direct trên máy in. Máy in sẽ phát ra tín hiệu Wi-Fi riêng (giống như một điểm truy cập).
- Trên điện thoại, người dùng vào phần cài đặt Wi-Fi và chọn kết nối với tên Wi-Fi của máy in.
- Sau khi kết nối, người dùng mở tài liệu cần in và chọn “In” → máy in sẽ xuất hiện trong danh sách thiết bị có sẵn.

- Thực hiện in ấn trực tiếp từ điện thoại đến máy in mà không cần bộ định tuyến Wi-Fi trung gian.

b) Giao tiếp gần (NFC)

NFC (Near Field Communication) là một công nghệ kết nối không dây tầm ngắn đầy tiềm năng, đặc biệt trong bối cảnh phát triển mạnh mẽ của các thiết bị di động và hệ thống thanh toán thông minh. NFC cho phép hai thiết bị trao đổi dữ liệu khi tiếp xúc gần nhau, thường trong phạm vi dưới 4 cm, giúp việc kết nối trở nên nhanh chóng, đơn giản và an toàn.

Hình 9.6. Ứng dụng NFC trong đọc thông tin thẻ căn cước công dân

Một điểm nổi bật của NFC là khả năng thiết lập kết nối tức thì mà không cần cấu hình phức tạp, rất phù hợp trong các tình huống như thanh toán không tiếp xúc (contactless payment), chia sẻ danh bạ, hoặc ghép nối nhanh giữa điện thoại và loa Bluetooth. Công nghệ này hoạt động dựa trên nguyên lý cảm ứng từ trường giữa các ăng-ten tích hợp trong thiết bị, cho phép truyền dữ liệu tốc độ thấp nhưng ổn định và tiết kiệm năng lượng. NFC được áp dụng nhiều trong internet vạn vật (IoT)

c) Bluetooth tiết kiệm năng lượng (BLE) / Bluetooth 5

Bluetooth Low Energy (BLE), còn gọi là Bluetooth Smart, là một phiên bản của giao thức Bluetooth được thiết kế đặc biệt để tối ưu hóa mức tiêu thụ năng lượng trong khi vẫn duy trì khả năng kết nối không dây hiệu quả. BLE là lựa chọn lý tưởng cho các thiết bị IoT như cảm biến, vòng tay theo dõi sức khỏe, thiết bị y tế, và các thiết bị đeo thông minh, nơi mà pin nhỏ và thời gian hoạt động kéo dài là yếu tố quan trọng.

Khác với Bluetooth truyền thống (còn gọi là Bluetooth Classic), BLE hoạt động theo cơ chế tạm thời đánh thức - truyền dữ liệu - ngủ, nhờ đó thiết bị có thể hoạt động trong nhiều tháng, thậm chí nhiều năm chỉ với một viên pin cúc áo. Bên cạnh đó, BLE sử dụng các gói tin ngắn, truyền tải nhanh và hiệu quả, phù hợp cho các ứng dụng trao đổi dữ liệu có dung lượng nhỏ nhưng cần thường xuyên và ổn định.

Bluetooth 5, được giới thiệu vào năm 2016, là một bản nâng cấp quan trọng của BLE, mang lại nhiều cải tiến vượt trội về phạm vi, tốc độ và độ tin cậy của kết nối. Cụ thể, Bluetooth 5 hỗ trợ phạm vi truyền dữ liệu xa gấp 4 lần, tốc độ nhanh gấp 2 lần, và công suất truyền dữ liệu tăng gấp 8 lần so với Bluetooth 4.2. Nhờ đó, Bluetooth 5 không chỉ cải thiện hiệu suất trong môi trường nhiều nhiễu mà còn mở rộng khả năng triển khai cho các ứng dụng công nghiệp, nhà thông minh và theo dõi vị trí trong thời gian thực.

BLE được áp dụng rất nhiều trong các dự án phát triển ứng dụng trên nền tảng Arduino, Raspberry Pi. Một số thiết bị nhúng sử dụng module BLE như HM-10, nRF52, hoặc ESP32.

LEVEL:3.3V
JDY-3x
Power:3.6V—6V
STATE
RXD
TXD
GND
VCC
EN

Hình 9.7. Module Bluetooth 5.0 BLE MESH JDY-24M

Bluetooth Low Energy và Bluetooth 5 đang trở thành tiêu chuẩn không thể thiếu trong thiết kế hệ thống nhúng hiện đại. Hiểu rõ cơ chế hoạt động và tiềm năng ứng dụng của BLE sẽ trang bị cho sinh viên nền tảng vững chắc để xây dựng các giải pháp thông minh, tiết kiệm năng lượng và sẵn sàng cho các mô hình sản phẩm thực tế trong công nghiệp và đời sống.

d) Zigbee

Zigbee là một giao thức truyền thông không dây dựa trên tiêu chuẩn IEEE 802.15.4, được thiết kế đặc biệt cho các ứng dụng có tốc độ dữ liệu thấp, tiêu thụ năng lượng thấp và yêu cầu tính ổn định cao trong môi trường mạng nhiều thiết bị. Zigbee là một công nghệ cốt lõi trong các hệ thống mạng cảm biến không dây (WSN), nhà thông minh (Smart Home) và các ứng dụng IoT công nghiệp.

Hình 9.8. Module Thu Phát Zigbee CC2530 UART 2.4G DL-22

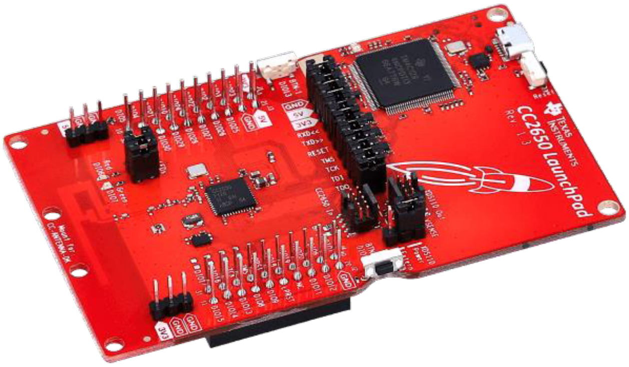
Một trong những ưu điểm nổi bật của Zigbee là khả năng hỗ trợ mạng dạng lưới (mesh network), giúp mở rộng vùng phủ sóng và tăng độ tin cậy của hệ thống nhờ khả năng định tuyến linh hoạt giữa các nút mạng. Zigbee có thể hoạt động trong các dải tần 2.4 GHz, 868 MHz hoặc 915 MHz tùy theo khu vực, với tốc độ truyền dữ liệu từ 20 kbps đến 250 kbps - đủ để truyền các gói dữ liệu nhỏ như trạng thái cảm biến, lệnh điều khiển thiết bị, v.v.

Zigbee có kiến trúc mạng bao gồm ba loại thiết bị: Coordinator (thiết bị điều phối), Router (bộ định tuyến), và End Device (thiết bị đầu cuối). Mô hình này cho phép hệ thống mở rộng linh hoạt, tiết kiệm năng lượng cho các thiết bị đầu cuối, đồng thời vẫn duy trì được kết nối ổn định trong toàn bộ mạng. Điều này rất phù hợp với các ứng dụng yêu cầu triển khai số lượng lớn cảm biến hoặc thiết bị đầu cuối trong môi trường phức tạp như nhà máy, nông trại, hoặc các tòa nhà lớn.

Trong bối cảnh chuyển đổi số và phát triển các đô thị thông minh, Zigbee tiếp tục là một trong những lựa chọn hàng đầu cho các hệ thống IoT quy mô vừa và nhỏ. Việc nắm vững kiến thức về Zigbee sẽ giúp sinh viên không chỉ hiểu công nghệ mà còn sẵn sàng phát triển các ứng dụng thực tiễn có giá trị trong công nghiệp, nông nghiệp, y tế và dân dụng.

e) 6LoWPAN

6LoWPAN là một giao thức mạng được thiết kế để cho phép giao tiếp IPv6 qua các mạng không dây cá nhân công suất thấp (Low Power Wireless Personal Area Networks - LoWPANs), đặc biệt là các mạng sử dụng chuẩn IEEE 802.15.4 như Zigbee. Với tư cách là một giảng viên đại học và chuyên gia mạng, tôi đánh giá 6LoWPAN là một cầu nối quan trọng giữa thế giới Internet truyền thống (IP) và các thiết bị nhúng công suất thấp trong môi trường IoT.



Hình 9.9. Modul TI CC2650

Một trong những thách thức lớn khi tích hợp các thiết bị nhúng vào Internet là dung lượng bộ nhớ và năng lượng hạn chế, trong khi giao thức IPv6 lại có kích thước gói tin lớn và phức tạp. 6LoWPAN giải quyết vấn đề này bằng cách nén tiêu đề gói IPv6, phân mảnh gói tin, và hỗ trợ định tuyến trong mạng không dây dạng lưới, giúp các thiết bị nhỏ gọn vẫn có thể giao tiếp trực tiếp với Internet mà không cần cổng trung gian chuyển đổi giao thức.

Khác với các giao thức như Zigbee sử dụng giao thức ứng dụng riêng biệt, 6LoWPAN cho phép các thiết bị nhúng sử dụng giao thức IP chuẩn để gửi và nhận dữ liệu. Điều này mang lại lợi thế rất lớn về khả năng mở rộng, tương thích, và tích hợp hệ thống. Các thiết bị cảm biến, điều khiển hoặc giám sát có thể kết nối trực tiếp đến máy chủ web, ứng dụng di động, hoặc các dịch vụ đám mây mà không cần hạ tầng phức tạp.

f) LTE

LTE (Long Term Evolution) là một tiêu chuẩn mạng di động tiên tiến thuộc thế hệ thứ tư (4G), được thiết kế nhằm cung cấp tốc độ truyền dữ liệu cao, độ trễ thấp, và hiệu suất phổ tần tối ưu cho các thiết bị di động. LTE không chỉ đơn thuần là sự cải tiến của 3G mà còn là bước chuyển mình quan trọng trong việc xây dựng một nền tảng viễn thông hiện đại, phù hợp với nhu cầu ứng dụng dữ liệu di động, dịch vụ truyền phát video, và Internet vạn vật (IoT) ngày nay.

Về mặt kỹ thuật, LTE sử dụng OFDM (Orthogonal Frequency Division Multiplexing) ở đường xuống (downlink) và SC-FDMA (Single Carrier-FDMA) ở đường lên (uplink), giúp cải thiện khả năng chống nhiễu và tăng hiệu suất truyền dẫn trong môi trường nhiều người dùng. LTE hỗ trợ tốc độ lên tới 100 Mbps ở downlink và 50 Mbps ở uplink, đồng thời có độ trễ chỉ khoảng vài mili-giây - một yếu tố rất quan trọng cho các ứng dụng thời gian thực như gọi video hoặc điều khiển từ xa.

LTE không chỉ đơn thuần là một công nghệ truyền dữ liệu nhanh hơn mà còn là một hệ sinh thái mạng bao gồm kiến trúc Evolved Packet Core (EPC), trạm gốc eNodeB, và các giao thức điều khiển như S1, X2, tạo nên một kiến trúc mạng thuần IP (All-IP Network). Điều này giúp LTE tích hợp tốt hơn với các dịch vụ Internet hiện đại và dễ dàng mở rộng lên các thế hệ mạng tiếp theo như LTE-Advanced và 5G NR.

Trong bối cảnh phát triển IoT và các hệ thống điều khiển thông minh, LTE còn đóng vai trò như một lớp truyền dẫn ổn định và có độ phủ sóng rộng. Các phiên bản mở rộng như LTE-M (LTE for Machines) và NB-IoT (Narrowband IoT) được thiết kế đặc biệt cho thiết bị nhúng tiêu thụ năng lượng thấp, cho phép các hệ thống cảm biến và điều khiển có thể hoạt động lâu dài với chi phí thấp và mức độ tin cậy cao.

g) 5G

5G viết tắt của Fifth Generation Mobile Network, là thế hệ mạng di động mới nhất, kế thừa và vượt trội hơn hẳn so với các chuẩn mạng trước đó như 3G và 4G LTE. 5G không chỉ đơn thuần là cải tiến về tốc độ truyền dữ liệu mà còn mở ra một hệ sinh thái kết nối thông minh, phục vụ hàng tỷ thiết bị - từ điện thoại, thiết bị IoT đến xe tự hành và các hệ thống điều khiển công nghiệp thông minh. Tốc độ truyền dữ liệu có thể đạt đến 10 Gbps, độ trễ chỉ 1 mili-giây, và khả năng kết nối lên tới 1 triệu thiết bị/km^2, biến 5G trở thành nền tảng hạ tầng số thiết yếu cho kỷ nguyên công nghiệp 4.0.

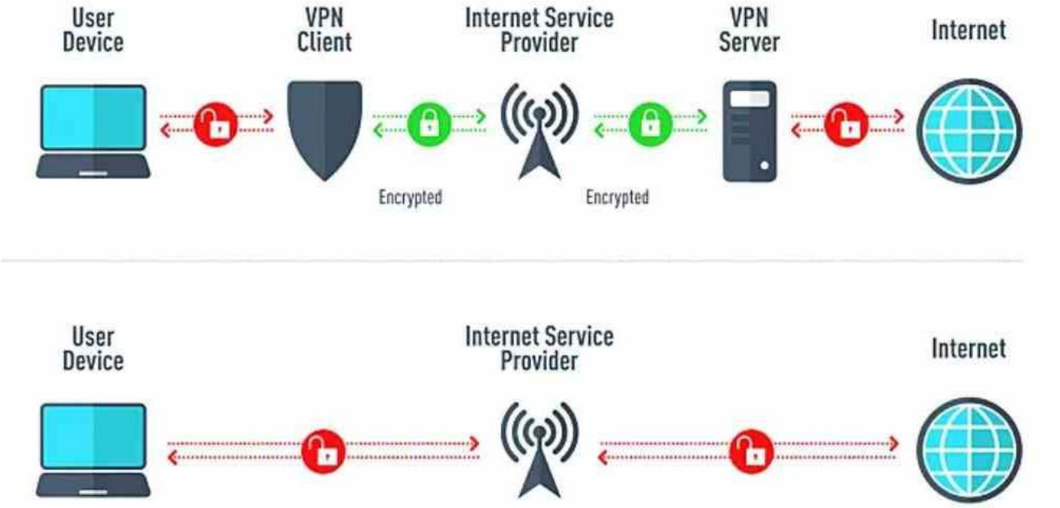
Từ góc độ kỹ thuật, 5G dựa trên nhiều công nghệ tiên tiến như sóng mmWave, beamforming, massive MIMO (Multiple Input Multiple Output), và network slicing. Nhờ đó, mạng 5G có thể tùy biến và phân chia tài nguyên để phục vụ nhiều loại dịch vụ khác nhau - từ truyền tải video 8K, thực tế ảo (VR), đến truyền thông siêu tin cậy và thời gian thực cho y tế từ xa hoặc xe tự lái. Việc sử dụng kiến trúc ảo hóa và điện toán biên (edge computing) trong 5G cũng giúp giảm đáng kể độ trễ và tăng khả năng xử lý phân tán tại gần người dùng. Các thành phần cấu trúc của hệ thống 5G như: Trạm gốc gNodeB, mạng lõi 5GC (5G Core), và các giao thức điều khiển đi kèm.

Mặc dù còn nhiều công nghệ mạng tương lai khác tồn tại và đang được sử dụng, cuốn giáo trình lựa chọn giới thiệu tổng quan một số công nghệ mạng mang tính xu hướng công nghệ, việc tìm hiểu các công nghệ mạng này sẽ giúp sinh viên hình dung rõ ràng các ứng dụng thực tiễn trong thời đại IoT bùng nổ như nhà máy thông minh, thành phố thông minh, và hệ thống giao thông thông minh.

9.2. Thực hành - Sử dụng mạng riêng ảo

Trong phần thực hành này, sinh viên sẽ được tiếp cận và thao tác với dịch vụ mạng riêng ảo. Đây là công nghệ hỗ trợ truy cập mạng hiệu quả hơn, đảm bảo tính riêng tư, vượt qua giới hạn truy cập và góp phần tăng cường bảo mật.

Mạng riêng ảo (Virtual Private Network - VPN) là công nghệ cho phép thiết lập một kết nối mạng an toàn và mã hóa giữa thiết bị người dùng và một máy chủ trung gian. Nhờ đó, toàn bộ dữ liệu được truyền qua VPN sẽ được bảo vệ khỏi việc bị theo dõi hoặc thu thập, đặc biệt hữu ích khi truy cập Internet qua các mạng Wi-Fi công cộng. Một ưu điểm quan trọng khác của VPN là khả năng ẩn địa chỉ IP thực và giả lập vị trí địa lý, cho phép người dùng truy cập vào các dịch vụ bị giới hạn theo quốc gia.

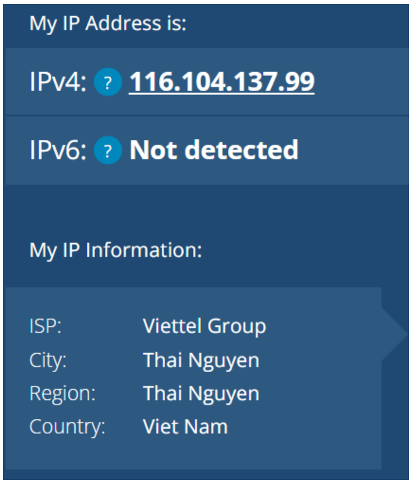


Hình 9.10. Truy cập Internet với VPN (trên) và không có VPN (dưới)

Khi sử dụng VPN, toàn bộ lưu lượng mạng từ thiết bị đều đi qua kết nối VPN. Điều này giúp bảo vệ không chỉ trình duyệt mà cả các ứng dụng khác như email, trò chơi trực tuyến hay phần mềm quản lý tệp. Trong thực tế, người dùng cá nhân thường sử dụng VPN để đảm bảo quyền riêng tư, truy cập nội dung bị chặn theo khu vực hoặc kết nối an toàn tới mạng nội bộ của doanh nghiệp khi làm việc từ xa.

Trong bài thực hành này, sinh viên sẽ được hướng dẫn sử dụng dịch vụ ProtonVPN, một nhà cung cấp VPN phổ biến với gói miễn phí chất lượng tốt. Sau khi kết nối thành công, sinh viên sẽ quan sát sự thay đổi về địa chỉ IP.

Trước khi cài đặt và sử dụng ProtonVPN, hãy truy cập vào web site `https://whatismyipaddress.com/` để xem thông tin về địa chỉ IP public hiện tại.



My IP Address is:

IPv4: ? **116.104.137.99**

IPv6: ? **Not detected**

My IP Information:

ISP: Viettel Group
City: Thai Nguyen
Region: Thai Nguyen
Country: Viet Nam

Hình 9.11. Thông tin về địa chỉ IP public của thiết bị

Bước 1: Truy cập https://protonvpn.com, đăng ký tài khoản miễn phí.

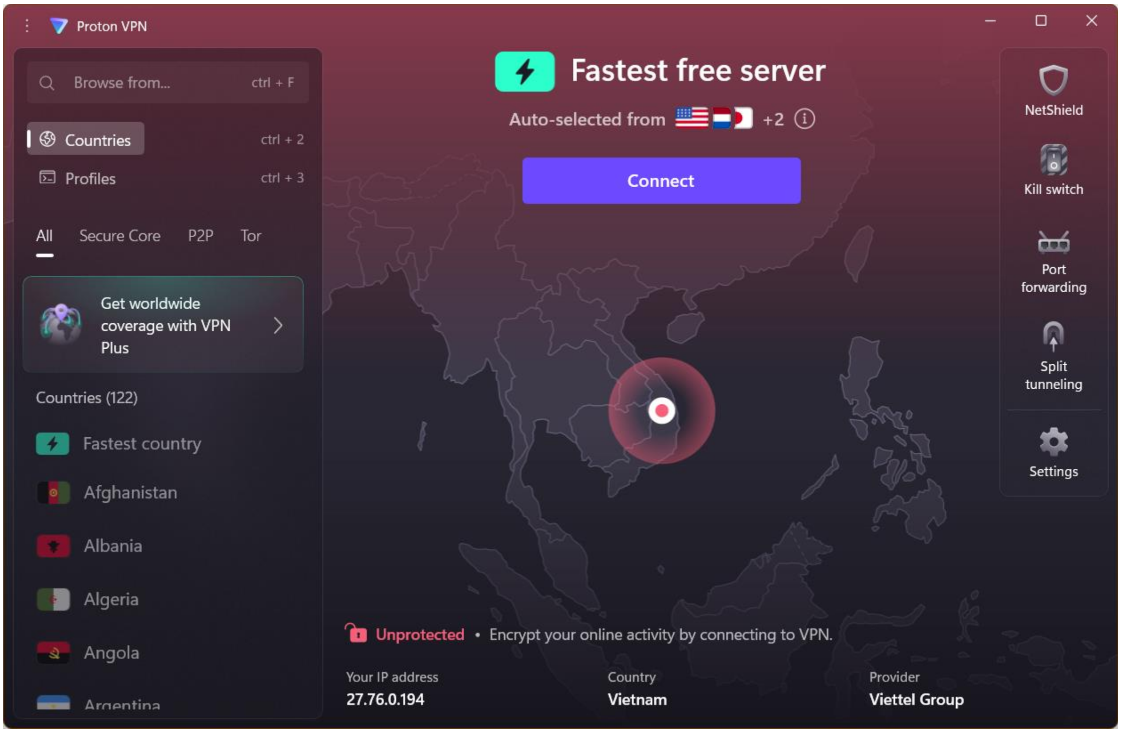
ProtonVPN là một dịch vụ trả phí. Tuy nhiên, ProtonVPN cũng cung cấp gói dịch vụ miễn phí. Gói miễn phí cho phép sử dụng trên 1 thiết bị và có thể kết nối ngẫu nhiên tới một trong 4 máy chủ ở 4 quốc gia khác nhau. Nếu cần sử dụng trên nhiều thiết bị hoặc kết nối tới máy chủ ở những quốc gia khác cần đăng ký gói trả phí.

Bước 2: Tải và cài đặt phần mềm ProtonVPN cho Windows.

Để sử dụng dịch vụ VPN của Proton cần cài đặt phần mềm VPN client. Phần mềm này được cung cấp sẵn trên website của ProtonVPN. Chú ý lựa chọn phiên bản phù hợp với hệ điều hành (Windows, Mac, Linux) và kiến trúc vi xử lý (ARM, x64, x86). Thực hiện cài đặt phần mềm này như những phần mềm thông thường.

Bước 3: Đăng nhập và kết nối

Sau khi cài đặt, hãy đăng nhập vào phần mềm sử dụng tài khoản đã đăng ký ở bước 1.


Proton VPN
Browse from... ctrl + F
Countries ctrl + 2
Profiles ctrl + 3
All
Secure Core
P2P
Tor
Get worldwide coverage with VPN Plus
Countries (122)
Fastest country
Afghanistan
Albania
Algeria
Angola
Argentina
Fastest free server
Connect
NetShield
Kill switch
Port forwarding
Split tunneling
Settings

Auto-selected from +2

Unprotected • Encrypt your online activity by connecting to VPN.

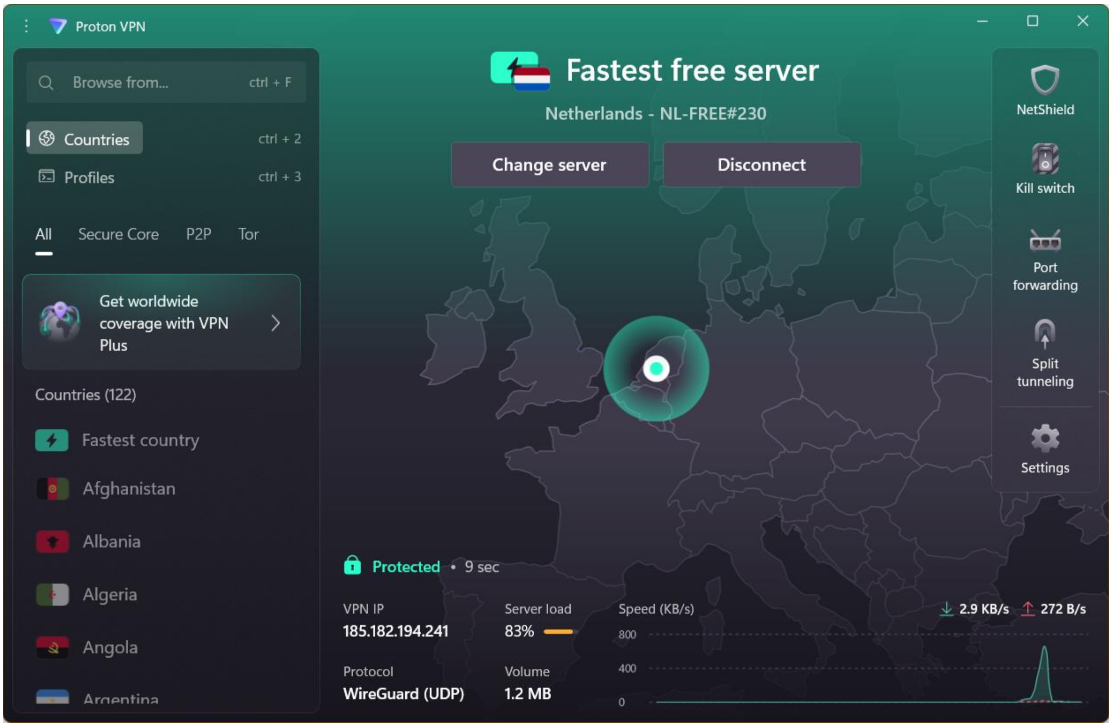
Your IP address: 27.76.0.194

Country: Vietnam

Provider: Viettel Group

Hình 9.12. Giao diện ProtonVPN client sau khi đăng nhập

Bấm nút Connect để kết nối tới Server. ProtonVPN client sẽ tự động lựa chọn một trong số các server miễn phí có tốc độ tốt nhất để kết nối.


Proton VPN
Browse from... ctrl + F
Countries ctrl + 2
Profiles ctrl + 3
All
Secure Core
P2P
Tor
Get worldwide coverage with VPN Plus
Countries (122)
Fastest country
Afghanistan
Albania
Algeria
Angola
Argentina
Fastest free server
Change server
Disconnect
NetShield
Kill switch
Port forwarding
Split tunneling
Settings
Speed (KB/s)
800
400
0

Netherlands - NL-FREE#230

Protected • 9 sec

VPN IP: 185.182.194.241

Server load: 83%

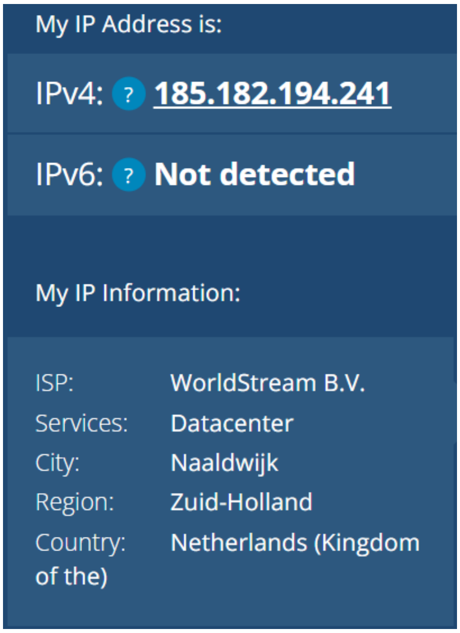
Protocol: WireGuard (UDP)

Volume: 1.2 MB

2.9 KB/s 272 B/s

Hình 9.13. Giao diện ProtonVPN sau khi kết nối tới server

Nếu mở lại trang https://whatismyip.com chúng ta sẽ thấy thông tin về địa chỉ IP và vùng địa lý đã hoàn toàn thay đổi. Toàn Internet giờ đây sẽ xem như thiết bị của bạn đang ở Hà Lan. Vì vậy, bạn có thể truy cập được vào website vốn bị cấm ở Việt Nam.



My IP Address is:

IPv4: 185.182.194.241

IPv6: Not detected

My IP Information:

ISP: WorldStream B.V.
Services: Datacenter
City: Naaldwijk
Region: Zuid-Holland
Country: Netherlands (Kingdom of the)

Hình 9.14. Thông tin mới về kết nối Internet sau khi dùng VPN

Sau khi thực hiện xong kết nối VPN, tất cả các thiết bị trên Internet sẽ nhìn nhận thiết bị của bạn có địa chỉ IP public của máy chủ VPN, chứ không nhìn thấy địa chỉ IP public thực sự của thiết bị nữa. Nói cách khác, từ giờ bạn sẽ truy cập vào Internet với vai trò của máy chủ VPN đang kết nối tới. Vì vậy, bạn có thể vượt qua các rào cản do ISP hoặc máy chủ khác đặt ra.

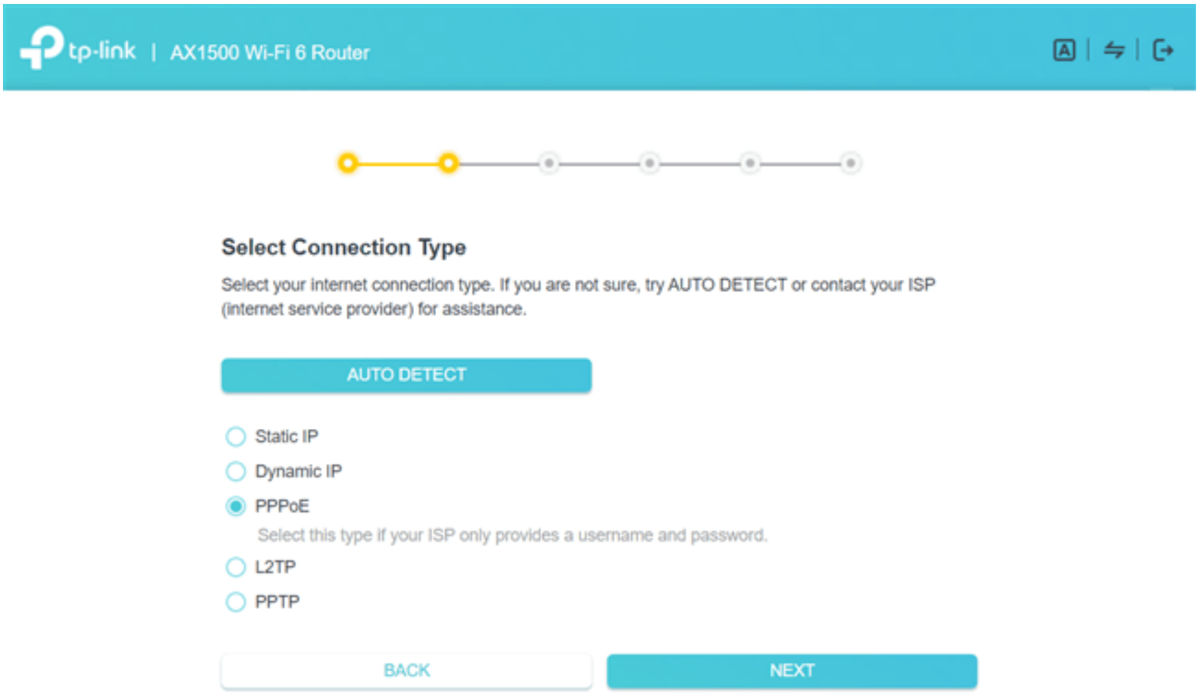
BÀI TẬP VÀ CÂU HỎI ÔN TẬP

Bài tập thực hành

Bài thực hành này giúp sinh viên làm quen với cách cấu hình PPPoE để kết nối tới ISP trên thiết bị router của gia đình. Thông tin cấu hình thường đã được ISP (Viettel, FPT, VNPT, etc.) cung cấp khi đăng ký sử dụng dịch vụ.

Bài thực hành này lấy thiết bị router TP-Link WiFi 6 làm thiết bị thực hành. Hầu hết các thiết bị ONT gia dụng hiện nay đều cung cấp giao diện cấu hình trên web tương tự nhau. Sinh viên có thể cấu hình trên thiết bị khác hoặc sử dụng phần mềm mô phỏng như packet tracer. Sinh viên tự thực hiện trên thiết bị ở gia đình.

Bước 1: Chọn giao thức PPPoE


tp-link | AX1500 Wi-Fi 6 Router

Select Connection Type

Select your internet connection type. If you are not sure, try AUTO DETECT or contact your ISP (internet service provider) for assistance.

AUTO DETECT

○ Static IP

○ Dynamic IP

● PPPoE

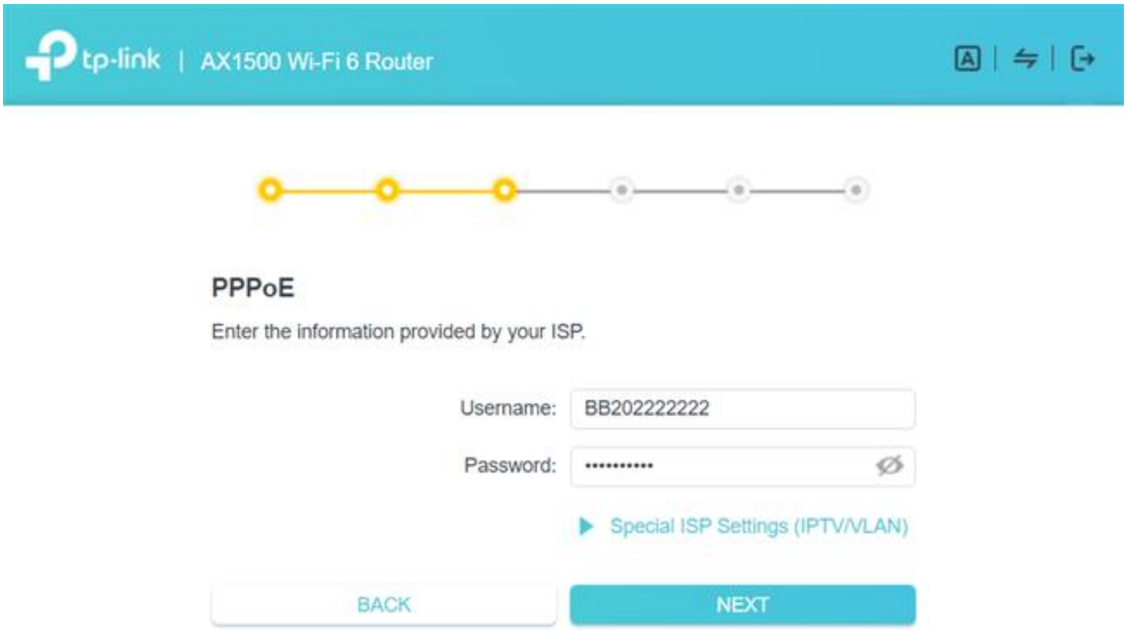
Select this type if your ISP only provides a username and password.

○ L2TP

○ PPTP

BACK
NEXT

Bước 2: Nhập tên và mật khẩu người dùng


tp-link | AX1500 Wi-Fi 6 Router

PPPoE

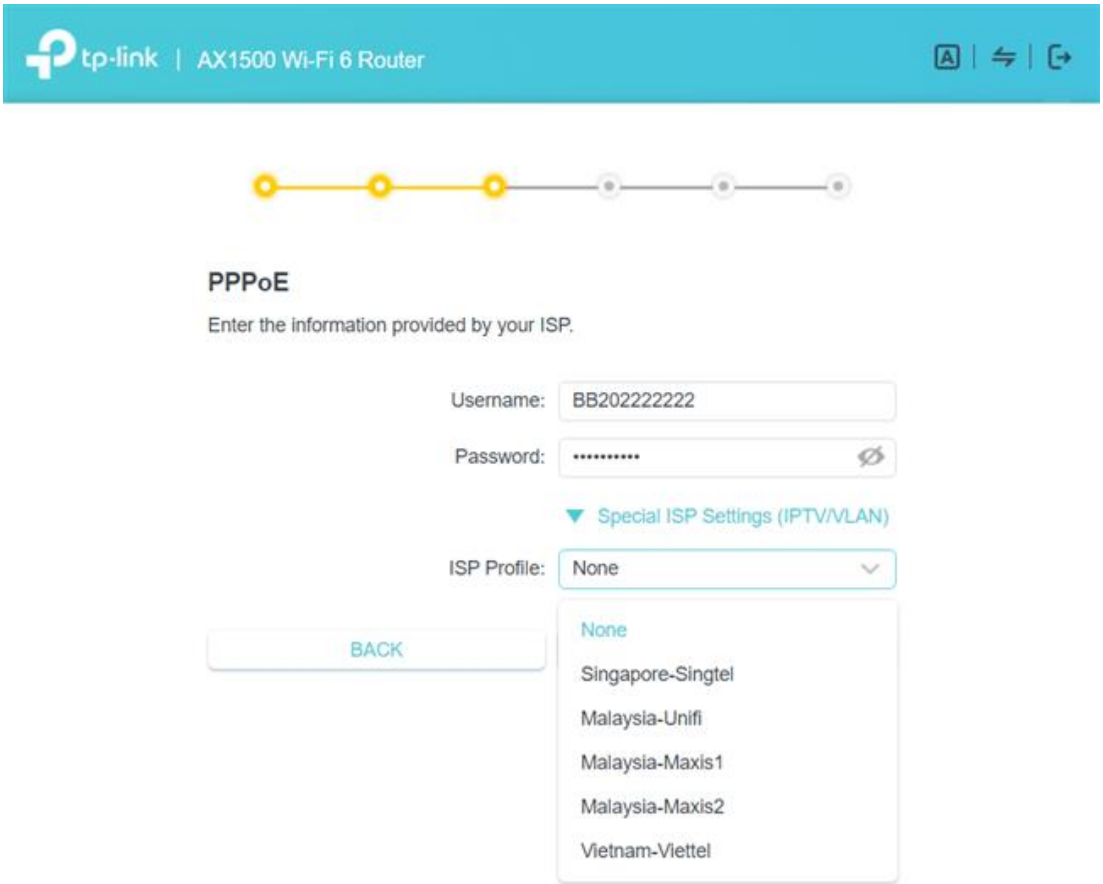
Enter the information provided by your ISP.

Username: BB202222222

Password: ••••••••••

Special ISP Settings (IPTV/VLAN)
BACK
NEXT

Bước 3: Nếu nhà cung cấp dịch vụ có sử dụng cấu hình tùy chỉnh thì click vào Spacial ISP Setting (IPTV/VLAN) và chọn tùy chỉnh phù hợp.


tp-link | AX1500 Wi-Fi 6 Router
Special ISP Settings (IPTV/VLAN)
BACK
None
Singapore-Singtel
Malaysia-Unifi
Malaysia-Maxis1
Malaysia-Maxis2
Vietnam-Viettel

PPPoE

Enter the information provided by your ISP.

Username: BB202222222

Password: ••••••••••

ISP Profile: None

Câu hỏi lý thuyết

1) Nếu bạn cần thiết kế mạng cho một nhà xưởng lớn có cả thiết bị di động và thiết bị cố định, bạn sẽ chọn kết hợp công nghệ nào giữa Ethernet và Wi-Fi? Giải thích vì sao.
2) Giả sử bạn cần truyền dữ liệu có dung lượng lớn, độ trễ thấp giữa các chi nhánh ngân hàng ở hai thành phố. Vì sao nên ưu tiên sử dụng công nghệ WAN nào?
3) Hãy so sánh đặc điểm hoạt động giữa mạng LAN sử dụng Wi-Fi 6 và mạng WAN sử dụng công nghệ FTTH về tốc độ, độ ổn định và khả năng mở rộng.
4) Nếu một hộ gia đình gặp hiện tượng mất kết nối khi quá nhiều thiết bị cùng dùng Wi-Fi, đâu có thể là nguyên nhân về mặt công nghệ? Bạn sẽ đề xuất giải pháp gì?
5) Khi cấu hình PPPoE trong mạng gia đình, vì sao cần phải có tên đăng nhập và mật khẩu xác thực? Điều này hỗ trợ ISP như thế nào trong quản lý người dùng?
6) Vì sao công nghệ Zigbee và Bluetooth BLE lại phù hợp với các hệ thống cảm biến trong nhà thông minh hơn là Wi-Fi hoặc LTE?
7) Nếu bạn cần thiết kế một hệ thống truyền dữ liệu từ các cảm biến nhiệt độ đến máy chủ mỗi 5 phút và yêu cầu pin cảm biến kéo dài trên 1 năm, bạn chọn công nghệ mạng nào? Giải thích lựa chọn của bạn.

8) Trong môi trường đô thị đông đúc và nhiều nhiễu sóng, bạn cần truyền video HD trực tiếp giữa 2 thiết bị di động. Bạn sẽ chọn giữa 5G và Wi-Fi Direct? Vì sao?
9) So sánh cấu trúc mạng dựa trên PPPoE và DHCP. Tình huống nào thì PPPoE thích hợp hơn?
10) Một công ty khởi nghiệp định phát triển ứng dụng IoT theo dõi xe rác đô thị, với điều kiện truyền ít dữ liệu, chi phí thấp và vùng phủ rộng. Nên sử dụng LTE-M, Zigbee, hay 6LoWPAN? Vì sao?

TÀI LIỆU THAM KHẢO

[1] A. S. Tanenbaum and D. Wetherall, *Computer Networks*, 6th ed. Boston, MA, USA: Pearson, 2021.

[2] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 9th ed. Boston, MA, USA: Pearson, 2023.

[3] B. Ciubotaru and G.-M. Muntean, *Advanced Network Programming: Principles and Techniques*. Berlin, Germany: Springer, 2013.

[4] K. R. Fall and W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2025.

[5] I. Griffiths, *Programming C# 10: Build Cloud, Web, and Desktop Applications*. Sebastopol, CA, USA: O’Reilly Media, 2022.

[6] W. Stallings, *Data and Computer Communications*, 10th ed. Boston, MA, USA: Pearson, 2017.

StarTech
APC

ISBN: 978-604-76-3150-6

9 786047 631506

Giá: 152.000 đ